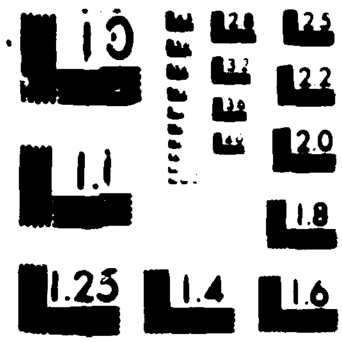
AD-A181 739 BMPAT· A PROGRAM FOR CALCULATION AND DISPLAY OF THE RESPONSE OF THREE-DIM (U) DEFENCE RESEARCH ESTABLISHMENT ATLANTIC DARTMOUTH (NOVA SCOTI 1/1

UNCLASSIFIED A J COLLIER APR 87 DREA-TC-87/306 F/G 12/5 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

DTIC FILE COPY

(3)

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense Nationale
Bureau de Recherche et Développement

TECHNICAL COMMUNICATION 87/306
April 1987

AD-A181 739

BMPAT: A PROGRAM FOR CALCULATION AND DISPLAY OF THE RESPONSE OF THREE-DIMENSIONAL ARRAYS

A.J. Collier

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC
ELECTE
JUN 25 1987
S D
D

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

Canada

87 6 24 047

DEFENCE RESEARCH ESTABLISHMENT ATLANTIC
9 GROVE STREET
P.O. BOX 1012
DARTMOUTH, N.S.
B2Y 3Z7

TELEPHONE
(902) 426-3100

CENTRE DE RECHERCHES POUR LA DÉFENSE ATLANTIQUE
9 GROVE STREET
C.P. 1012
DARTMOUTH, N.É.
B2Y 3Z7

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

BMPAT: A PROGRAM FOR CALCULATION AND DISPLAY OF THE RESPONSE OF THREE-DIMENSIONAL ARRAYS

A.J. Collier

April 1987

Approved by H.M. Merklinger H/Surveillance Acoustics Section

DISTRIBUTION APPROVED BY R.F. Brown

D/UAD

TECHNICAL COMMUNICATION 87/306

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

Canada

ABSTRACT

The program BMPAT is a FORTRAN 77 program designed to readily evaluate the amplitude and phase response of general three-dimensional sonar receiving arrays to plane wave arrivals. The program output is directed to two output files, one for amplitude response and the other for phase response, both of which are formatted for plotting using the DREA Surveillance Acoustics Plotting Package, SAPLOT. Features in the program include the ability to handle array elements that have directional response, provision for element amplitude and phase imbalance and for array distortion. The beamforming operation is a linear operation and the steering vectors can either be input directly or can be calculated for phase-to-a-plane beamforming. There are several amplitude window functions resident in the program and additional functions can be input via tables. Beam patterns are sampled in planes which can be arbitrarily oriented relative to the array co-ordinate system. Beamwidths in the sample planes are calculated and the directivity index of beams can optionally be calculated.

SOMMAIRE

Le logiciel BMPAT, rédigé en FORTRAN 77, est conçu pour l'évaluation rapide de la réponse en amplitude et de la réponse en phase de réseaux récepteurs sonar tridimensionnels généraux en présence d'ondes planes d'arrivée. Le logiciel produit deux fichiers de sortie, l'un pour la réponse en amplitude et l'autre pour la réponse en phase; ces deux fichiers sont formatés en prévision du traçage au moyen de SAPLOT, le progiciel de traçage en surveillance acoustique de CRDA. Parmi les fonctions du logiciel, on compte la possibilité du traitement des éléments de réseau à réponse directive; de plus, le logiciel peut tenir compte de l'amplitude et de l'asymétrie de l'élément, ainsi que de la déformation du réseau. L'opération de conformation du faisceau est une opération linéaire; les vecteurs de pilotage peuvent être calculés en fonction de la conformation de faisceau de phase à plan. Le logiciel comporte plusieurs fonctions à fenêtre d'amplitude, tandis que d'autres fonctions peuvent être introduites au moyen de certains tableaux. Les diagrammes de faisceau sont échantillonnés dans des plans à orientation arbitraire par rapport au système de coordonnées du réseau. La largeur de faisceau dans les plans échantillonnés est calculée et il existe une option de calcul de l'indice de directivité des faisceaux.

TABLE OF CONTENTS

DTIC COPY INSPECTED 1

Codes
Dist | Avail and/or Special
A-1

LIST OF FIGURES

(Note: captions appear again under diagrams)

1. INTRODUCTION

It is often useful to be able to quickly evaluate and display the amplitude and phase response of sonar receiving arrays. The program BMPAT is designed for that purpose. It has been written to be as flexible as possible, both from the perspective of the array and beamforming operations and with respect to the user's interaction with the program. The program is written in FORTRAN 77 to operate on a VAX computer. Program output is directed to two output files, one for amplitude response and the other for phase response, formatted for display using the DREA Surveillance Acoustics Plotting Package, SAPLOT[1]. In this form the data is available for display and as ASCII data. Beam widths are calculated and the directivity index of beams can be calculated if desired. These quantities are output with the amplitude response.

The program architecture allows the use of the keyboard and of files for the input of data, and the flexibility to switch back and forth at will. This is an advantage when, for instance, a fairly complicated array is to be considered and it is desired to examine the amplitude response at a number of frequencies and for a number of beam steering directions. In such a case it would be reasonable to input the array element positions from a file, but the frequencies and beam steering directions could be readily input from the keyboard.

An attempt has been made to allow the maximum flexibility in the design of the array with a minimum number of restrictions. The number of elements allowed is a function only of the maximum number of elements specified by the program parameter MXNE. This number is currently set at 100 but can readily be increased. The element positions are completely unrestricted. The array elements can be directional, both for amplitude and phase response. Each element is permitted to take on one of four possible directional characteristics; these include omnidirectional, dipole and cardioid directivities as well as a user specified directivity as defined in a table. The element directivity is restricted to be rotationally invariant about the element axis; however the element axis can be arbitrarily oriented. The program allows for the inclusion of array distortion and for error in the amplitude and phase response of individual elements.

The beamforming operation is restricted to linear beamforming defined by the following expression:

$$B(\phi_L,\theta_L) = V^H S(\phi_L,\theta_L), \tag{1-1}$$

where ϕ_L,θ_L are the azimuthal and polar angles defining the look direction,

$B(\phi_L,\theta_L)$ is the complex beam output in the look direction,

$S(\phi_L,\theta_L)$ is the array signal vector (a column vector) for a plane wave arriving from the look direction,

and V^H is the complex steering vector (a row vector), where the superscript H represents Hermitian transpose.

In the program the look direction ϕ_L,θ_L is replaced by only the azimuthal angle, ϕ_L, as the beam response can only be examined in a plane. The sample plane however can be selectively oriented.

The array response that is calculated is the response to incoming plane waves that are assumed to be completely coherent over the array aperture. Response to other forms of wavefronts can be evaluated to a limited degree by utilizing the ability to use different element positions to calculate steering vectors, by utilizing the ability to specify element amplitude and phase response error or by inputting a steering vector directly that compensates for the non-plane nature of the expected arrivals.

The steering vector $\mathbf{V}^{\mathbf{H}}$ can either be input directly or can be calculated assuming phase to a plane beamforming. To beamform to a plane, only the steering direction is required for input, although the calculated vector can be modified by amplitude windowing. Other beamforming schemes require that the vector be input directly. For the calculated vectors a number of amplitude shading windows can be applied before beamforming. The program allows for rectangular, raised cosine, raised cosine squared, Hann, Hamming, Gaussian and Taylor windows as well as two table defined windows. The amplitude shading can be performed along a vertical axis, along a horizontal axis or over a plane, and two window functions can be simultaneously applied. There is also an option to compensate the window shading functions for unequal distribution of elements over the array aperture. In the calculation of steering vectors it is possible to calculate the vector either for the exact element positions or to define an array of assumed element positions from which the steering vectors are calculated.

2. DETAILED FUNCTIONAL DESCRIPTION

Program control is exercised through menu input with the following selectable options:

1 - Select keyboard for input
2 - Select a file for data input
3 - Enter frequency and sound speed
4 - Enter array element co-ordinates and element types
5 - Enter ideal (undistorted) array geometry
6 - Enter a table defining element response
7 - Enter element amplitude and phase imbalances
8 - Enter a steering vector
9 - Define amplitude windowing
10 - Calculate phase to a plane steering vector
11 - Calculate beam amplitude, phase response and/or directivity index
12 - Open output files
13 - Exit

The program has been designed so that it can be run interactively by responding to the keyboard prompts with little or no reference to this document. However, to avoid excessive keyboard input with the associated probability of typing errors, input files can be used, in which case the descriptions of the required input data contained in this document will be of value. It should also be noted that many of the functional steps require data that is input during the execution of other program steps. Thus it is important to use care in selecting the order of the functional steps. In the event that program operation is being controlled from the keyboard the user is prompted for the required steps; however, if program control is from a file then errors of this type may be fatal. In some cases the program will proceed, but default values will be used for required data that have not been entered. In such cases the user is informed.

2.1 Option 1 - Keyboard input

On entry to the program the keyboard is the input data source, but if at some point a file has been selected for data input, selecting step 1 from the menu returns control to the keyboard. Thus the use of the menu to select keyboard or file input allows a great deal of flexibility in inputting data.

2.2 Option 2 - File input

On selection of menu function 2 the next required input (from the currently active input source) is the name of the new input file. If there is an error in opening this file the user will be prompted again for the file name if the keyboard is the active input, but if the input source is a file,

the program stops. For all input files the first line MUST be a line of text containing from 0 to 80 characters. Subsequent data in the file must contain menu selections which control the function of the program, with each menu selection followed immediately by the required input data.

2.3 Option 3 - Entry of frequency and sound speed

This step requires only two lines of input: line 1 containing the frequency and line 2 the speed of sound. The speed of sound must be in the same units as those used in specifying the array geometry. It has a default option of 1500 m/s, selected by entering a carriage return rather than a value.

2.4 Option 4 - Entry of Array geometry

In program step 4 the array co-ordinate system for input of the array geometry is specified as a Cartesian, cylindrical or spherical co-ordinate system, the number of array elements is input, the element positions are input and the directionality of each element is defined.

2.4.1 Array co-ordinate system

The array co-ordinate system is defined as a left-hand Cartesian co-ordinate system, which can be imagined to have the x and y-axes horizontal with the x-axis pointing north, the y-axis east, and the z-axis vertically up. The option is provided to specify element postions in this Cartesian system or in either a cylindrical or a polar co-ordinate system. If the cylindrical system is selected, the x-variable corresponds to the radius, the y-variable to the azimuthal angle (positive angles relative to the x-axis (north) are to the east) and the z-variable to the distance along the z-axis. For the polar system, the x-variable corresponds to the radius, the y-variable to the azimuthal angle and the z-variable to the polar angle which is POSITIVE above the horizontal and NEGATIVE below. (NOTE that this convention for polar angles is not the same as would be expected with a left-hand co-ordinate system where the polar angle is equivalent to left-hand rotation about the y-axis.) C selects Cartesian input co-ordinates, Y cylindrical co-ordinates and P polar co-ordinates. If cylindical or polar co-ordinates are used for input they are converted to Cartesian immediately. The co-ordinate units must be consistent with the units used for the sound speed which is input in step 3. All angular co-ordinates are in decimal degrees.

2.4.2 Array element types

The element type defines the phase and amplitude response of the elements. The response of the elements can only be varied as a function of the polar angle (ie: the element response is rotationally invariant), but the maximum response direction for each element in the array can be oriented in any direction. The following selections of element type are available:

Type 0 - omnidirectional element,

1 - element response is table defined. If this option is selected it is necessary to input the element response table by choosing step 4 from the menu. If the element response table has not been input at the time of the beam response calculation (step 11) the elements are assumed to be omnidirectional.

2 - dipole element. The amplitude response = abs(cos α) and the phase response = 0° for $\alpha \le 90^\circ$

= 180° for $\alpha > 90^\circ$ deg where α is the angle relative to the element axis

3 - cardioid element. The amplitude response = $.5 + .5\cos(\alpha)$ and the phase response = 0°

2.4.3 Input for step 4

Following is a description of the input data for step 4:

line 1 - C,Y or P (lower case letters are accepted) specifying input co-ordinate system to be Cartesian, cylindical or polar.
line 2 - an integer specifying the number of elements in the array.
line 3 - for element 1; the element co-ordinates, element type and the azimuthal, polar angle defining the orientation of the element axis. (The element type must be an integer number)
line 4 - for element 2; ....
line 5 - for element 3; ....
etc.

2.4.4 Example

This paragraph contains a sample of input to the program which selects step 4 and defines a circular array of 6 elements, of which three are cardioid and three are dipole elements, with the peak element reponses oriented in the radial direction (see figure 1). Descriptive comments, which are not part of the input are italicized.

4	*step 4*
Y	*cylindrical co-ordinates*
6	*6 elements in array*
10.,0.,0.,3,0.,0. *	*element 1 -- co-ords (10 units, 0°,0 units),type 3, axis (0°, 0°)* **
10.,60.,0.,2,60.,0.	*element 2*
10.,120.,0.,3,120.,0.	*element 3*
10.,180.,0.,2,180.,0.	*element 4*
10.,240.,0.,3,240.,0.	*element 5*
10.,300.,0.,2,300.,0.	*element 6*

---

* In this and all following examples integer and real inputs are distinguished from each other by inclusion of the decimal point in the real inputs; however, in the program free format is used for input of numerical data and thus the decimal points need not be included. Commas are used as the delimiter between variables in all of the examples but spaces are also acceptable.

** The units in which the element positions are defined are the same as those used to define the speed of sound, which is a required input in program step 3.

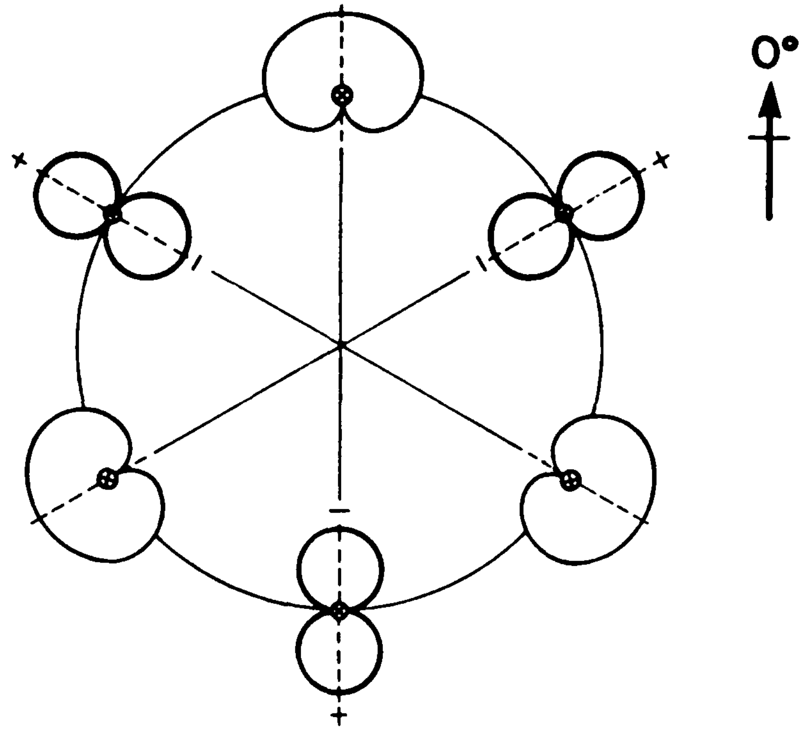


Figure 1. Schematic diagram of a 6 element circular array with three cardioid and three dipole elements. The element positions are shown, as well as the element amplitude responses and the element axes.

2.5 Option 5 - Entry of ideal (undistorted) array geometry

In the case of a distorted array the actual and ideal positions of array elements may not be the same. In calculating steering weights for beamforming the ideal element positions may optionally be used, but for array response the element signals depend on the actual positions. Thus this functional step allows the response of a distorted array to be examined for cases in which the steering vectors are calculated. If steering vectors are input directly then the ideal array geomety is not required.

The required input format for this step of the program is as follows:

line 1	- C,Y or P specifying the co-ordinate system to be Cartesian, cylindrical or polar
line 2	- element 1 positional co-ordinates
line 3	- element 2 positional co-ordinates
.	
.	
line N+1	- element N positional co-ordinates

If a distorted version of the circular array shown in figure 1 was input in step 4 and it is desired to define the undistorted array for the purpose of calculating the steering vectors, the program input would be as follows:

5	*step 5*
Y	*cylindrical co-ordinates*
10.,0.,0.,3,0.,0. [1]	*element 1 co-ordinates (10 units, 0°,0 units),*
10.,60.,0.,2,60.,0.	*element 2 co-ordinates*
10.,120.,0.,3,120.,0.	*element 3 co-ordinates*
10.,180.,0.,2,180.,0.	*element 4 co-ordinates*
10.,240.,0.,3,240.,0.	*element 5 co-ordinates*
10.,300.,0.,2,300.,0.	*element 6 co-ordinates*

2.6 Option 6 - Entry of element response tables

To handle element responses other than omnidirectional, dipole and cardioid, the element directionality must be input via a table. Element response is linearly interpolated from this table during the calculation of array beam response for the appropriate array elements. As currently configured, only a single element response table is handled by the program. Tables consist of the independent variable, which is the angle in degrees relative to the element axis. This variable must increase monotonically and usually ranges over values from 0 to 180°. Associated with each independent variable is the element amplitude response in dB and the phase response in degrees. Phase values must fall in the range ± 360°. Following is an example of step 6 input to define a dipole element response, although due to the limited number of points included in the example, the response is a fairly crude approximation. Note that table entry is terminated by a -1.

6	*step 6*
0.,0.,0.	*angle re axis, amplitude response in dB, phase response*
20.,-0.5,0.	*Note that the independent variable and 2 associated dependent*
40.,-2.3,0.	*variables must be input in a single line*
60.,-3.0,0.	
80.,-15.2,0.	
89.9,-30.,0.	
90.1,-30.,180.	
100.,-15.2,180.	
120,-3.0,180.	
140.,-2.3,180.	
160.,-0.5,180.	
180.,0.,180.	
-1	*table entry is terminated*

2.7 Option 7 - Entry of element amplitude and phase imbalance

It is through this step that element amplitude and phase errors can be applied. These errors can either be input or calculated. If the input option is selected, amplitude and phase error for each element are required. Calculated errors are taken from uniform random distributions and the only input required is the maximum amplitude and phase errors (plus a seed for the random number generator). In addition to applying errors to array elements, one of the available options in this step is to reset previously specified errors to zero. Once element errors have been set they remain (even if a new array is defined) until new values are input or the errors are zeroed. Errors are initially set to zero.

The input required by step 7 is as follows:

line 1 - either **R** to indicate errors are to be random (this is the default)
I to indicate errors are to be input
or **Z** to indicate element errors to be zeroed.

	if line 1 is R	if line 1 is I	if line 1 is Z
line 2	integer seed for random number generator	amp(dB), phase(deg) error for element 1	no more input required
line 3	maximum amp(dB), phase(deg) error	amplitude,phase error for element 2	-
line 4	-	amplitude, phase error for element 3	-
.			
.	-	etc.	-

2.8 Option 8 - Entry of a steering vector

The steering vector is a complex vector of order equal to the number of array elements. The vector V^H in equation (1-1) is the steering vector. This vector can be input directly through step 8 or calculated for phase to a plane beamforming through step 10. The input for step 8 consists of n lines, where n is the number of array elements, each containing the real and imaginary parts of one steering vector element. These complex elements represent the amplitude and phase factor applied to each of the array elements. As an example the following would be the input for a two element array that is to be beamformed to generate a dipole response:

1., 0.
-1., 0.

2.9 Option 9 - Define amplitude windowing

Step 9 is the program function which enables amplitude weighting of the array before beamforming. The windowing operations that are defined in this step are implemented at the time of beamforming, which is performed in step 11. When the steering vector is input directly through step 8, the windows defined in step 9 will not be implemented at the time of beamforming. Note that once amplitude weighting has been set up through step 9, it will be applied until the weights are redefined or the flag enabling the windowing operation is turned off.

Within step 9 there are three functional choices, one of which is selected by the first line of input.

"I" in the first line selects input of a table defining an amplitude window,
"W" selects the definition of the window function and
"T" toggles the window flag (enables/disables windows).

The window flag is automatically raised when W is selected. If on entry to step 9, a window function has not been defined through a previous entry into step 9, selecting "T" has no effect since the window flag cannot be raised if no window has previously been defined. If the window flag is not raised at the time of beamforming all elements are assigned an amplitude weight of unity, which defines a rectangular window.

2.9.1 Window tables

Inputting a table allows any window amplitude function to be employed. The program can currently accept two tables, each defining an amplitude window, although only one table can be input for each selection of function 9. Thus in order to input two tables, 9 must be selected

twice. When these tables are employed the window amplitudes are interpolated from the appropriate table during the beamforming operation in step 11, using the interpolation subroutine TBINTP.

The assumptions concerning the definition of windows in tables are as follows:

i. Windows are symmetric
ii. the array aperture is 2, with the extreme elements positioned at -1 and 1
iii. in the tables the independent variable is the normalized aperture which must increase monotonically from 0 to 1 (only 1/2 window is defined due to assumed symmetry)
iv. the dependent variable is the associated amplitude.

2.9.2 Resident window functions

In addition to the two windows that can be defined in the window table, the following window functions are available:

i. rectangular weighting
ii. Hann(ing) weighting
iii. Hamming weighting
iv. raised cosine weighting
v. raised cosine squared weighting
vi. Gaussian weighting
vii. Taylor weighting.

These windows are calculated with no consideration for variation in the directional response of the elements. The functional form of each of these window is given below in terms of "p", the projection of the element position on the normalized window axis. p takes on values from -1 to 1.

For the Hann weighting function the quantity ξ, the number of half-wavelengths in the projected aperture, is also used in order to zero the window a half-wavelength beyond the aperture, rather than at the end elements. The Hann, Hamming and Gaussian windows are defined in Harris[2] and the Taylor window in Trenholm and Young[3].

i. The Hann weighting function is defined as,

$$WT = 0.5 + 0.5\cos(\kappa p \pi), \tag{2.9-1}$$

where $\kappa = \xi/(\xi+2)$;

ii. the Hamming function as,

$$WT = 0.46 + 0.54\cos(\kappa p \pi) ; \tag{2.9-2}$$

iii. the raised cosine function as,

$$WT = \sigma + (.5 - \sigma/2)\cos(p\pi) \tag{2.9-3}$$

where σ is the fractional height of the pedestal and is constrained to be in the range 0. to 1 ;

iv. the raised cosine squared function as,

$$WT = \sigma + (1 - \sigma)\cos^2(p\pi/2); \tag{2.9-4}$$

v. the Gaussian function as,

$$WT = \exp(-0.5(\eta p)^2) \qquad (2.9\text{-}5)$$

where η is the parameter which establishes the mainlobe width and sidelobe levels. Larger values will give a wider main beam and lower sidelobes [2];

vi. and the Taylor function as,

$$WT = I_0\,[\gamma\sqrt{1-p^2}\,] \qquad (2.9\text{-}6)$$

where γ is the Taylor parameter which, similar to η above, controls the mainlobe width and sidelobe levels [3].

2.9.3 Definition of the window axis or plane

There are several choices of axes or planes to which the window function may be applied. These choices are:

1. window axis is in the horizontal plane fixed in azimuth independent of the steering direction, referred to as a steering-independent horizontal window;
2. window axis is in the horizontal plane but is defined as perpendicular to a vector pointing in the steering direction, referred to as a steering-dependent horizontal window;
3. window axis is coincident with the z-axis, referred to as a steering-independent vertical window;
4. window axis is coincident with the z-axis of a co-ordinate system which has its x-axis coincident with a vector pointing in the steering direction and its y-axis in the horizontal plane, referred to as a steering-dependent vertical window;
5. a two-dimensional window defined in a plane independent of steering direction, referred to as a steering-independent 2-D window;
6. a two-dimensional window defined in a plane which is normal to a vector pointing in the steering direction, referred to as a steering-dependent 2-D window.

For selections 1 and 5 input is required to orient the window axis or plane.

2.9.4 Two-dimensional windows *

The two-dimensional window weight calculationsare based on the radial distance of each element from the center of mass of the array in the plane of the window. These radial distances are normalized by the projected array aperture, which for the 2-D windows is the maximum of the projected radial distances (relative the center of mass) for all the elements.

---

* Caution should be exercised in the use of the 2-D windows. The author is unsure of the effect of the use of the 2-D windows on beampatterns. The option has been included in the program to increase the program flexibility.

2.9.5 Simultaneous application of two windows

It is possible to apply two amplitude window functions simultaneously, in which case the amplitude weight will be the product of the weights for each of the two windows. An example of the utility of this option is given by a circular array of staves, where the staves consist of a number of independent elements. In such a case it is customary to apply a window to each of the staves and then to apply a window across the staves. If the staves are oriented along the z-axis such a weighting scheme can be realized by simultaneosly applying a vertical steering-independent window and a horizontal window.

2.9.6 Aperture sharing**

Another option for the calculation of the window weights is referred to as aperture sharing. Most window schemes are considered in the context of equally spaced arrays. A window designed to suppress sidelobes in an equi-spaced array may have quite a different effect when applied to an unequally spaced array. In some circumstances it may be desirable to modify the window weight for each element proportional to the share of aperture that the element occupies. The option to compensate window weights for unequal spacing of elements along the window axis, or in the case of 2-D windows, in the window plane, has been included. In the case of one-dimensional windows the weight assigned to each element is inversely proportional to the fraction of the aperture occupied by the element. The aperture share is evaluated by looking for the nearest neighbours and assigning 1/2 the aperture between the nearest neighbours to the element. In the event of co-located elements this share is divided by the number of coincident elements. For 2-D windows the same procedure is followed except the radial distance from the center of mass of the projected array is used in place of the projection on the window axis. Based on these radial distances, an area of the window plane is associated with each element, rather than a share of a linear aperture as is the case for 1-D windows. The weight associated with each element is proportional to the inverse of the window area occupied by the element.

2.9.7 Input to step 9 (windowing)

The input is shown using the dialogue included in the program. The computer output is in small font; user input is emboldened.

To enter a window table:

```
Definition of window amplitude functions
Enter    E to input a window table
         W to define windows
         T to enable/disable windows? <W > ??  E
Enter the # to be used to identify table (max=2)?  1 or 2
Enter pairs of coords defining window:   abcissa,ordinate
         where abcissa = normalized aperture (0 to 1) and ordinate = window amplitude
-1 terminates entry
X1,Y1
X2,Y2
   .
   .
-1
```

The maximum number of points in tables is set by the parameter MXTBEN and is currently 100. The abcissa values X1,X2.... must increase monotonically over the range 0. to 1.

** The aperture sharing option should be exercised with care. It is included to add flexibility to the program; however, the results may differ from those desired.

To define an amplitude window the computer dialogue and response is as follows:

Definition of window amplitude functions
Enter E to input a window table
W to define windows
T to enable/disable windows? <W > ?? **W**
Define the amplitude window to use in beamforming
Enter the # of independent windows to apply (max 2)? **1 or 2**
** Enter parameters defining window 1 **
Select the window type as follows:
enter 1 to select angle independent horizontal window
2 to select angle dependent horizontal window
3 to select angle independent vertical window
4 to select angle dependent vertical window
5 to select angle independent 2-d window
6 to select angle dependent 2-d window ? **1 to 6**

If the above entry is 1, selecting an angle-independent horizontal window then:
Enter the azimuthal angle defining the window axis? **0 to 360**

Or if the above entry is 6, selecting an angle-independent 2-D window then:
Enter azimuthal,polar angle defining normal to the window plane? **-360 to 360, -90 to 90**

Select the window form as follows:
enter 0 to select a rectangular window
1 to define window from table 1
2 to define window from table 2
3 to select a Hann window
4 to select a Hamming window
5 to select a raised cosine window
6 to select a raised cosine squared window
7 to select a Gaussian window
8 to select a Taylor window ? **0 to 8**

If the above entry is 5 or 6, selecting a raised cosine or cosine squared window then:
Raised cosine (cos squared) window. Enter fractional height of pedestal: **0.to 1.**

If the above entry is 7, selecting a Gaussian window then:
Gaussian window selected
Enter factor controlling mainlobe width/sidelobe levels? <2.5>?? η

Or if the above entry is 8, selecting a Taylor window then:
Taylor window selected
Enter Taylor parameter controlling mainlobe width/sidelobe levels? <4.>?? γ

Compensate window weights for unequal spacing of elements over the window aperture? (NOTE that this option should be used with caution!) Y/N? <N>?? **Y or N**

At this point the dialogue is repeated from the selection of the window type if the number of applied windows is 2.

To enable/disable (toggle) the window:

> Definition of window amplitude functions
> Enter E to input a window table
> W to define windows
> T to enable/disable windows? <W > ?? **T** .

2.10 Option 10 - Calculation of the steering vector

The steering vector can be calculated for phase-to-a-plane beamforming employing either the actual positions of the array elements or their ideal positions. The only input required to this step in the program is the steering direction and an indication that the actual or assumed (see sections 2.4 and 2.5) element positions are to be used in calculating the vector. Dialogue with the program during this step is as follows:

> Phase to a plane beam steering selected
> Enter the beam steering direction in the array co-ordinate system:
> azimuthal, polar angles (deg)? **ϕ,θ**
> Use actual(A) or ideal(I) element positions for steering? <A>?? **A or I**

The polar angle, θ, is positive above the horizontal plane and negative below. In the event that ideal element positions are specified for calculation of the steering vector, but no ideal element positions have been input through step 5, the actual positions are used instead.

2.11 Option 11 - Beam amplitude and phase response, and directivity index

In this step the plane in which the beam pattern is to be sampled is defined, the beam amplitude and phase responses are calculated in this plane, the amplitude response is normalized, the normalized amplitude and phase responses are output and, optionally, the directivity index is calculated and output. Additionally the beamwidth in the sample plane is determined and output to the beam amplitude output file.

2.11.1 Beam response normalization

The output beam response is always normalized relative to a peak response of 0 dB. This peak reference is determined by one of the following user-selectable procedures:

1. The beam amplitude response is normalized so that the peak amplitude in the output plot is 0 dB;
2. The beam amplitude response is normalized relative to the response in the beam steering direction (this option is not available if the steering vector has been input directly through step 8); or
3. The beam amplitude response is normalized relative to the absolute peak amplitude response in three dimensions.

The factor used to normalize the beam amplitude response is output to the amplitude response file so the absolute beam response is not lost.

If option 3 is selected the absolute peak response is found by a search over three dimensions. The angular step size used in this search is user selectable, with a default value of 0.5 degrees. Either the ideal array, which has no array distortion nor element imbalance, or the actual array, including both distortion and element imbalance, can be used for determining the peak response. Thus it is possible to readily display the effects of distortion and element imbalance on the array response. It should be noted that the search for the peak response can be time consuming, and thus the angular step size for the search should be selected to be no smaller than necessary.

2.11.2 Beamwidth

The beamwidth (full beamwidth to 3 dB down points) in the sample plane is determined (in degrees) and written to the output file during the execution of this step, provided that an appropriate peak occurs in the sampled portion of the beam response. The criterion used to define a peak is the presence of a peak, that is within 0.1 dB of the maximum response in the plane, that is situated between points that are 3 dB down, relative to the maximum response in the plane. In cases where more than one peak occurs, the beamwidth output is the width of the narrowest peak. The bearing of the peak associated with the output beamwidth is output along with the beamwidth. Note that this bearing coincides with the bearing of the first maximum encountered between the 3-dB-down points. No interpolation is employed to refine this estimate, and thus if a precise peak position is desired a small sample interval must be used. Linear interpolation is employed in calculating the three-dB-down points. The size of the sampling increment can thus potentially affect the accuracy of the calculated beamwidth. If high accuracy is required, this should be reflected in the sampling interval that is input. If no peaks are identified in the sampled output, the beamwidth and peak location contain asterisks.

2.11.3 Directivity index

The directivity index (DI) is defined as the ratio, expressed in dB, between the power received by an omnidirectional receiver in an isotropic noise field, and the power received on the beam in the same field. For this calcualtion the peak beam response must be scaled to equal the response of the omnidirectional receiver. Calculation of the DI involves the same calculations as those made in searching for the absolute peak, and the same angular step size is used. In calculating the DI the actual array, as opposed to the ideal array, is used. Calculation of the DI, just as searching for the absolute peak, is time consuming, and this should be kept in mind when selecting the angular step size for these calculations. The DI, when calculated, is written to the output amplitude file. If this output field contains asterisks it indicates that the DI has not been calculated.

2.11.4 Sampling the beam response

The beam responses are always sampled in a plane. Sampling is accomplished by calculating the response of the beam of interest, to plane waves arriving in the sample plane. This plane can be oriented in any direction. Orientation of the sample plane is accomplished through a series of rotational transformations, performed on the array co-ordinate system, which define a sample co-ordinate system. (These transformations are described in more detail in the following paragraphs.) The sample plane is the x-y plane in this sample co-ordinate system and the bearings in this plane are relative to the x-axis (the sample x-axis bears 0°).

Generally beam patterns are sampled only in the horizontal or vertical plane, and so the program is set up so that these sample planes are readily selected. If it is desired to sample the beam response in planes other than the horizontal or vertical, the required input is more complex. The co-ordinate transformations for horizontal/vertical sample planes are transparent to the user.

To sample the beam in the horizontal plane the only input required to set up the sample plane is the reference azimuth, the true bearing at which the sample bearing is 0°. For sampling in the vertical plane the required input is identical except the reference azimuth defines the orientation of the vertical plane. The sample bearing is always 0° in the horizontal plane with positive angles above the horizontal.

Although horizontal and vertical sampling is most usual, the user may wish to sample in other planes. One reason for sampling in other planes is to look at the effect of array tilt. Sampling in planes other than horizontal or vertical involves the use of co-ordinate transformations, as mentioned above. Both the array system and the sample system are left hand co-ordinate systems, as described in section 2.4. The sample system is defined relative to the array system by the sequence of rotations required to rotate the array system to be coincident with the sample system. This sequence involves a rotation about the original z-axis (ρ_z), a rotation about the new y-axis ($\rho_{y'}$) and finally another z-axis rotation ($\rho_{z''}$). All of these rotations have a left-handed sense which means they are positive in a clock-wise direction. The sequence of rotations to make the transformation from the array system to a sample system is shown schematically in figure 2.

In some circumstances it may be desired to either orient the x-axis of the sample system in the north plane of the array system or reference it to this plane. The $\rho_{z''}$ rotation required to put the x-axis in the north plane depends on the ρ_z and $\rho_{y'}$ rotations, and requires some calculation. Figure 3 illustrates the geometry of the situation. The angle δ is the $\rho_{z''}$ rotation which puts the x-axis of the beam system in the north plane. The expression for δ is as follows:

$$\delta = \text{sign} \cos^{-1}(\cos \rho_z \cos \rho_z \cos \gamma + \sin \rho_z \sin \gamma) \qquad (2.11\text{-}1)$$

where

$$\text{sign} = 1 \quad \text{for} \sin \rho_z \leq 0$$
$$= -1 \quad \text{for} \sin \rho_z > 0$$
$$\cos \gamma = (1 + \tan^2 \rho_{y'} \cos^2 \rho_z)^{-1/2}$$

and

$$\sin \gamma = \tan \rho_{y'} \cos \rho_z \cos \gamma .$$

Table 1 contains some examples of rotational coordinates that define sampling planes that are neither vertical nor horizontal.

Table 1. Examples of rotational transformations for defining non-vertical/horizontal beam sampling planes.

ρz	ρy'	ρz"	description of the sample plane
0	-45	0	sample plane is oriented 45° relative to the horizontal, sample system y-axis coincident with array system y-axis, sample system x-axis in north plane 45° above horizontal
30	-45	0	sample plane is oriented at 45° relative to the horizontal, sample system y-axis is horizontal bearing 120° relative to north, sample system x-axis is not in north plane and is 45° above horizontal
30	-45	-22.	sample plane is oriented at 45° relative to the horizontal, sample system x-axis is in the north plane, sample system y-axis is above the horizontal

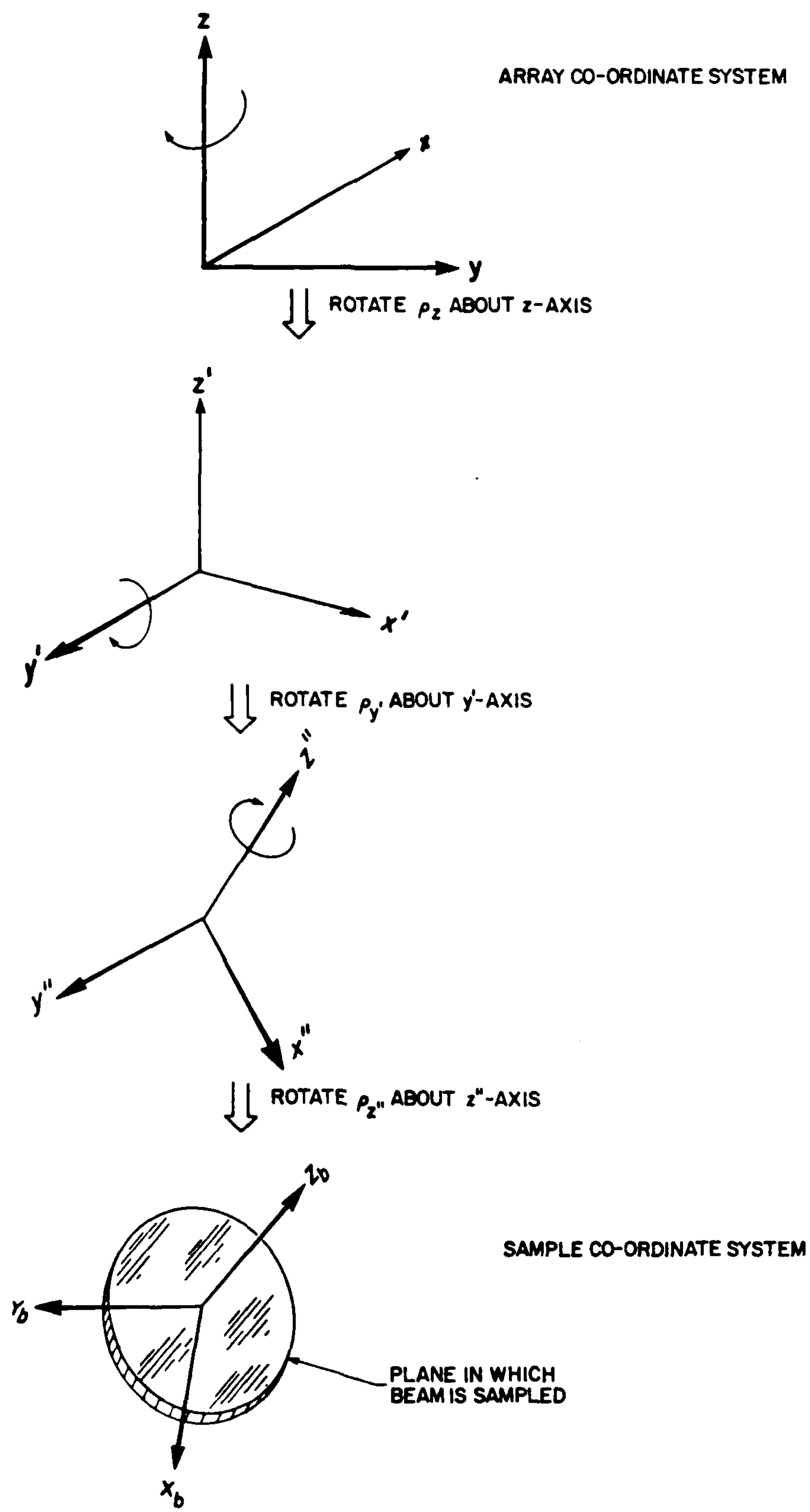


Figure 2. Rotations involved in transforming the array co-ordinate system to the sample co-ordinate system.

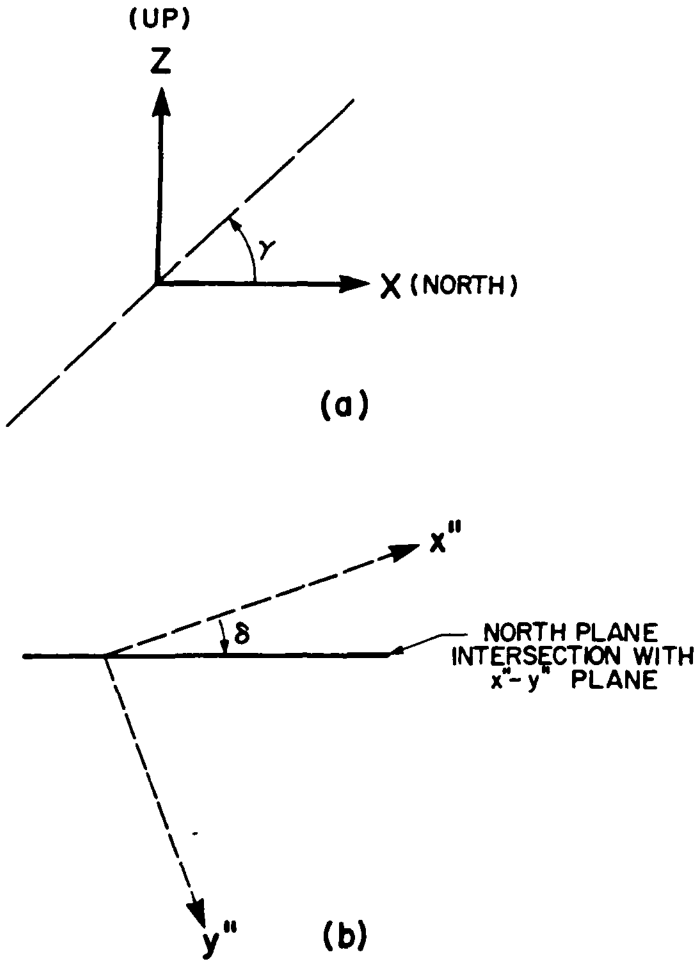


Figure 3. The rotation about the z"-axis required to put the x"-axis in the north plane is the angle δ illustrated in (b). δ is defined in terms of the angle γ, the inclination of the x"-y" plane in the north plane, shown in (a). The dashed line in (a) is the line along which the x"-y" plane cuts the north plane. Both δ and γ have a left-hand sense and thus the γ in this example will be negative and the δ positive.

2.11.5 Input for step 11

The required input to step 11 is demonstrated in the following input dialogue (the computer prompts are shown in small font and the possible responses are emboldened):

Is the beam to be sampled in the horizontal (H), vertical (V) or other (O) plane? <H>?? **H,V** or **O**

If the response to the above question is H indicating a horizontal sample plane then:

Enter the azimuthal angle to which the beam response is to be referenced? <0>?? **-360** to **360**

if the response is V indicating a vertical sample plane then:

Enter the azimuthal angle defining the orientation of the vertical sample plane ? <0>?? **-360** to **360**

or if the response is O indicating that the sample plane is neither horizontal nor vertical then:

Enter the z,y' rotations defining the sample plane? **-180 to 180, -90 to 90 .**

Now if the other (O) plane has been selected, if the active input is the keyboard, and the y' rotation entered above is in the range -85 to 85, THEN the user will be advised as to the z" rotation required to put the beam system x-axis in the north plane as follows:

With these z,y' rotations the z" rotation required to put the beam x-axis in the north plane is ...

The dialogue continues, independently of the output of this advisory.

Enter the z" rotation to orient the x-axis of the beam system in the x-y plane? **-360 to 360**

The dialogue now continues independently of the chosen sample plane:

Enter the angles at which the beam is to be sampled: min, max, increment? _,_,_
Enter the type of normalization to employ where,
1. sets the peak response to 0 dB
2. normalizes relative to the beam response in the steering direction
3. normalized relative to the peak 3-dimensional response (NOTE this is slow)
1,2 or 3? <1>?? **1,2 or 3**

If type 2 or 3 normalization is choosen then:

Use the response of the actual (A) or ideal (I) array for normalization? <A>?? **A or I**
(Be certain if selecting the ideal array that the ideal array geometry has been input through step 5)

The dialogue continues independently of the type of normalization:

Is the directivity index to be calculated - THIS IS TIME CONSUMING (Y/N)? <N>?? **Y or N**

Now if type 3 normalization has been choosen and/or the directivity index is to be calculated:

Enter angular step size for peak search/directivity calculation? <0.5>?? _

(Select this input carefully to minimize calculation time - step size should be as large as possible)

And the dialogue continues:

Enter the minimum beam power to plot:

(If a carriage return is entered the program will autoscale the output. This input only affects the range statement in the output SAPLOT file and does not modify the output data.)

2.12 Option 12 - Output file

New output files can be opened at any time by calling this functional step. The default output files, which are used until other output files are opened via step 12, are BMPAT.AMP and BMPAT.PHS. After step 12 has been executed the new output files will be FILENAME.AMP and FILENAME.PHS for the case where the new output file name has been specified as FILENAME.

The only dialogue with this step is as follows:

Enter name for output files (include no extension)? **FILENAME**

If there is an error in opening the new output file and the active input is the keyboard, the user will be prompted again. If the active input is a file the program will stop on this error.

2.13 Description of the contents of the plot information lines

There are three information lines output at the top of each plot generated by the program. The example following is an information header for an amplitude response plot. The information header output to the corresponding phase plot does not include the beamwidth, the directivity index nor the normalization information.

PLOT #2, FREQ = 100Hz, BMWIDTH=164.3deg at -33.deg, DI = -4.2 dB
NORM=1,NORM FACTOR= 24.8 dB, WINDOW(S) #=2 type,form 2,3 3,7
AMPLITUDE RESPONSE IN HORIZONTAL PLANE relative 0. degrees

Plot # — simply keeps track of the number of the plot in a single session with BMPAT. The # is incremented each time functional step 11 is called in the program. As soon as the program is exited the count is reset. Both the amplitude response plot and the corresponding phase response plot have the same plot #.

Beamwidth — information is only output to the amplitude response file. The beamwidth and location of the peak from which the beamwidth was determined is indicated. The peak location may not coincide exactly with the maximum response as a 0.1 dB tolerance is used in finding peaks. In the case of more than one peak in the plot, the beamwidth output to the header is the width of the narrowest peak. Asterisks are output in this field if there is no peak in the plot. (see section 2.11.2)

DI — The directivity index (DI) is output only on amplitude response plots. If calculation of the DI was selected at the time the plot was being generated (in step 11), then the value will be contained in this field; otherwise the field will contain asterisks.

Norm — indicates the the type of normalization used to scale the beam response, 1,2 or 3, defined as follows: (see also section 2.11.1)

1. indicates maximum response in the output plot is set to 0 dB;
2. indicates the amplitude response has been normalized relative to the beam response in the steering direction; (Either the actual or ideal array can be used to determine the normalizing power.)
3. indicates the maximum beam response in any direction is used as the normalizing power. (Again either the actual or ideal array can be used as the reference.)

Norm factor — is the amplitude factor, expressed in dB, used to normalize the output beam response. Output of this factor allows the user to recover the absolute beam response, since the beam responses that are output are relative. The normalization factor is only output on the amplitude response plots.

Windows (see section 2.9)

The second line of plot information indicates the amplitude windows applied during the beamforming. The above example indicates there were two windows applied, the first a type 2 window of form 3 and the second a type 3 window of form 7. If no windows are applied during beamforming then the type,form information does not appear.

The window TYPES are as follows:

1. angle-independent horizontal window
2. angle-dependent horizontal window
3. angle-independent vertical window
4. angle-dependent vertical window
5. angle-independent 2-D window
6. angle-dependent 2-D window.

The window FORMS are:

0. rectangular window
1. window from table 1
2. window from table 2
3. Hann window
4. Hamming window
5. raised cosine window
6. raised cosine squared window
7. Gaussian window
8. Taylor window.

If 10 is added to the window form it indicates that aperture averaging was used in calculating the amplitude window.

The third information line indicates whether the plot is an amplitude or phase response plot, and indicates the orientation of the plane in which the beam was sampled. (see section 2.11.4)

3. SOME EXAMPLES

This section contains several examples of beam responses obtained from the program. Only the plots for these are included in this section. Input dialogue, listings of the contents of input files, and output file listings have been included in the appendices.

3.1 Circular array of cardioid and dipole elements.

The array in this example is the circular array shown in figure 1. Figures 4 to 7 show the beam response for beams steered in direction 0,0. Figure 4 is the amplitude response in the horizontal plane at a frequency of 100 Hz (the sound speed in this example is input as 4800 ft/sec) and figure 5 is the corresponding phase response. Figure 6 is the vertical response for this same beam. The horizontal response at 1000 Hz is shown in figure 7. The input dialogue generating these plots is included in Appendix A, as is a listing of the input file CARD1.DAT in which the array geometry is defined, and an edited listing of the SAPLOT output file BMPAT.AMP.

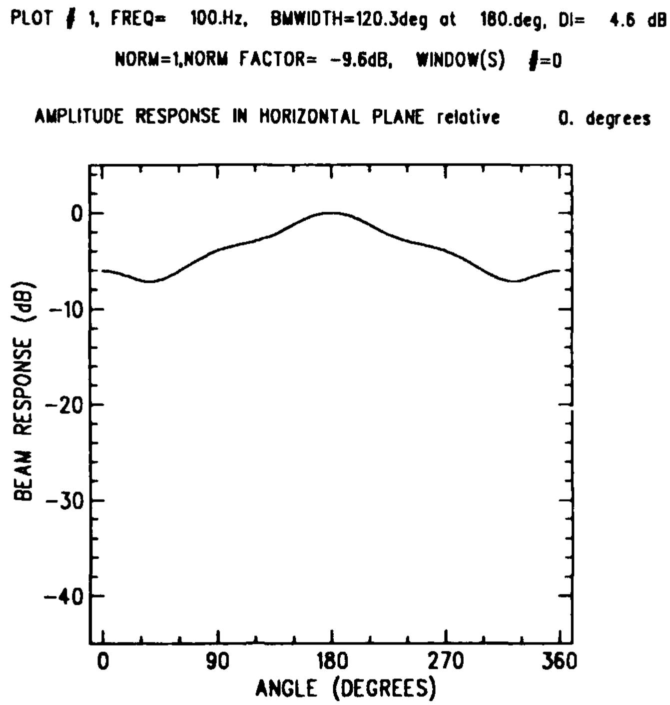


Figure 4. Amplitude response in the horizontal plane for the circular array of Figure 1. The steering direction is 0°,0° and the frequency 100 Hz. Note that the peak response is 180° off the steering direction.

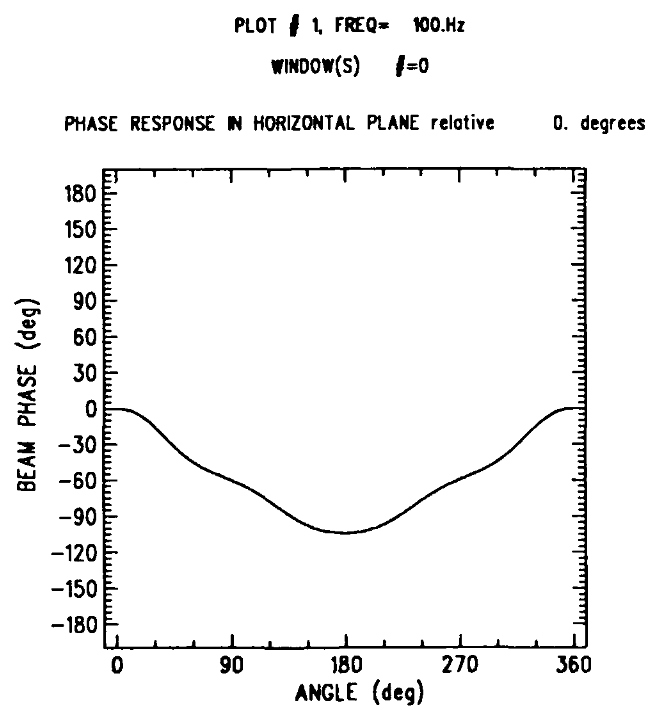


Figure 5. Phase response corresponding to Figure 4.

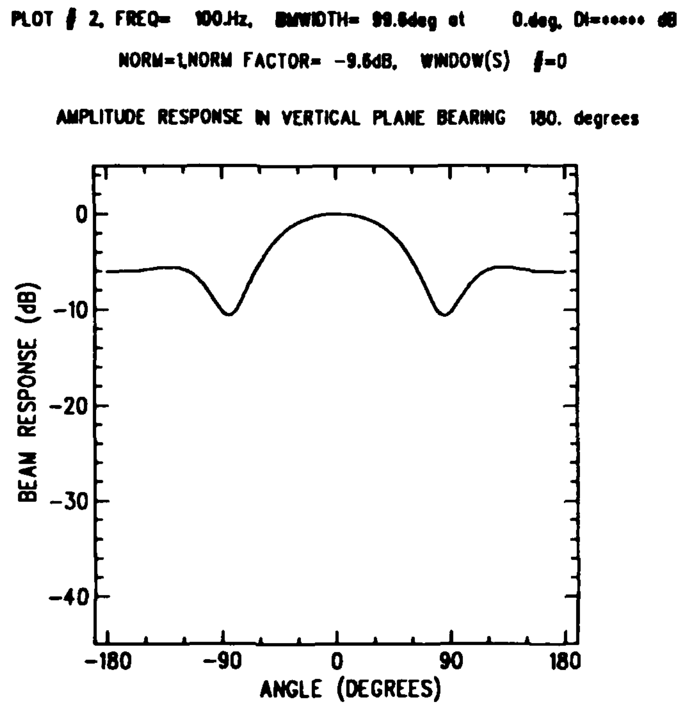


Figure 6. Amplitude response in the vertical plane for the circular array of figure 1. The steering direction is 0°,0° and the frequency 100 Hz. The directivity index was not calculated and thus this field in the output header is filled with asterisks.

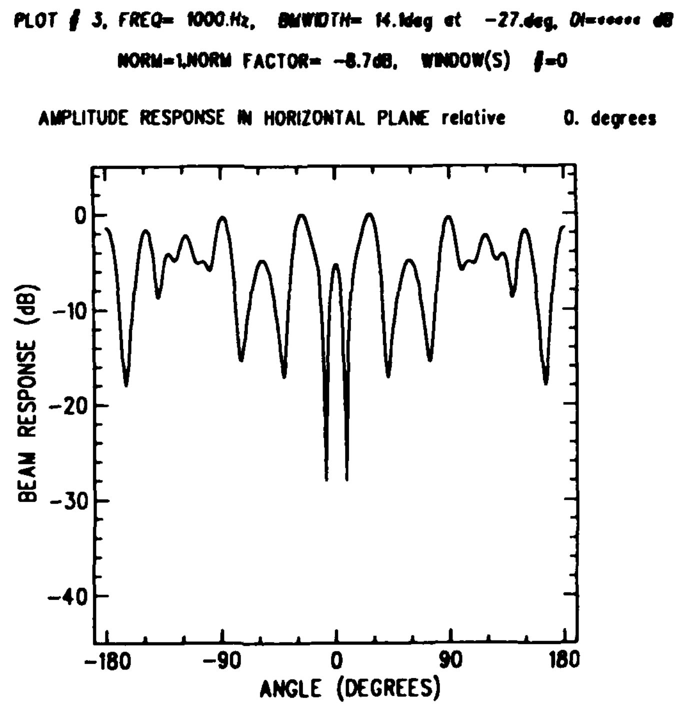


Figure 7. Amplitude response in the horizontal plane for the circular array of Figure 1. The steering direction is 0°,0° and the frequency 1000 Hz. The output beamwidth is calculated from the narrowest of the several peaks, which is at 0°.

3.2 Seventeen element line array

The second example array is a line array of 17 equispaced omnidirectional elements "cut" for 50 Hz (half wavelenth separation of the elements at 50 Hz). Figures 8 to 12 show some responses for this array. The listing of the input file in which the array geometry was defined and the dialogue from which these responses were generated is included in Appendix B.

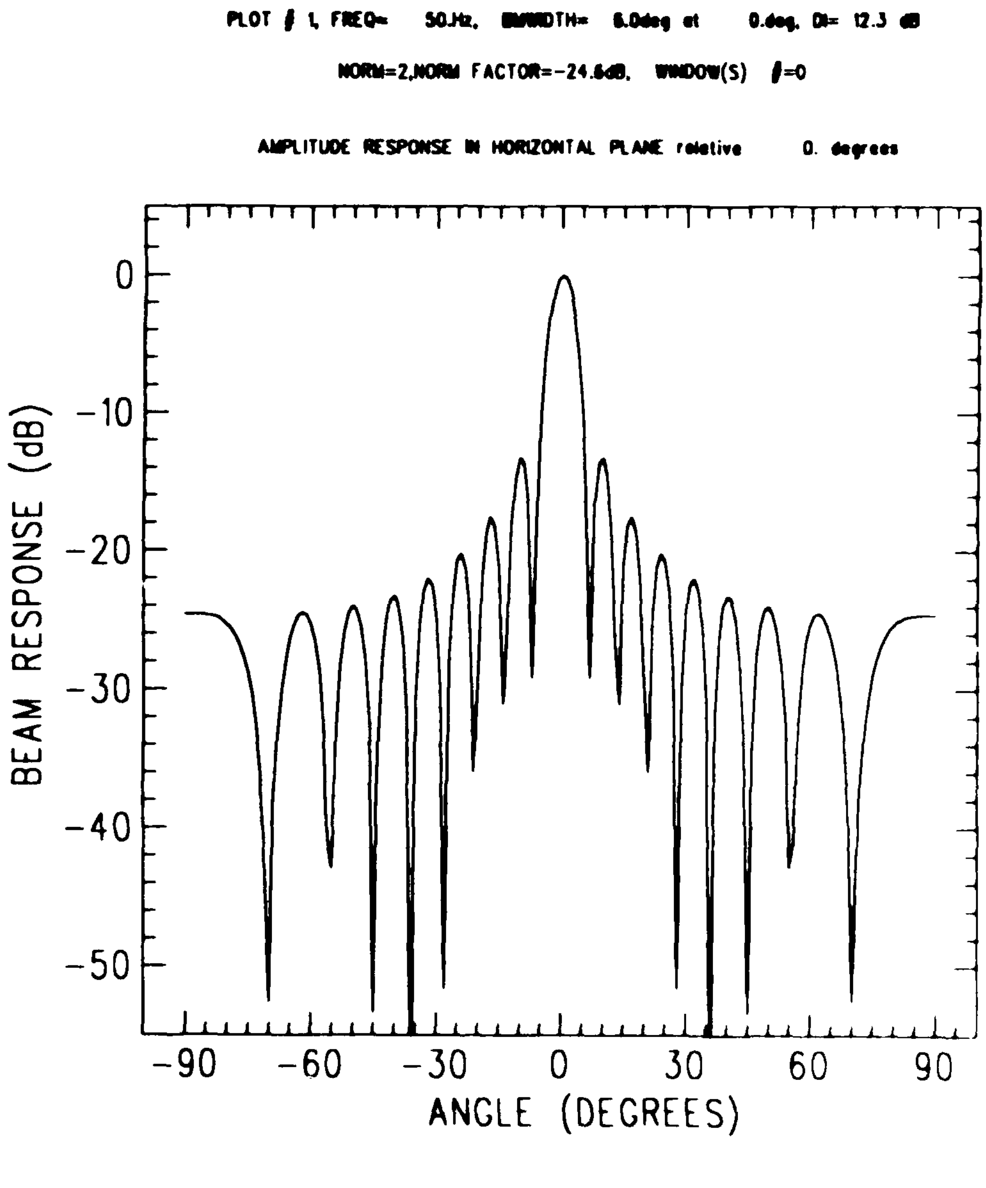


Figure 8. Amplitude response of the broadside beam of a 17 element horizontal line array at 50 Hz. The beam is sampled in the horizontal plane. No amplitude window has been applied before beamforming.

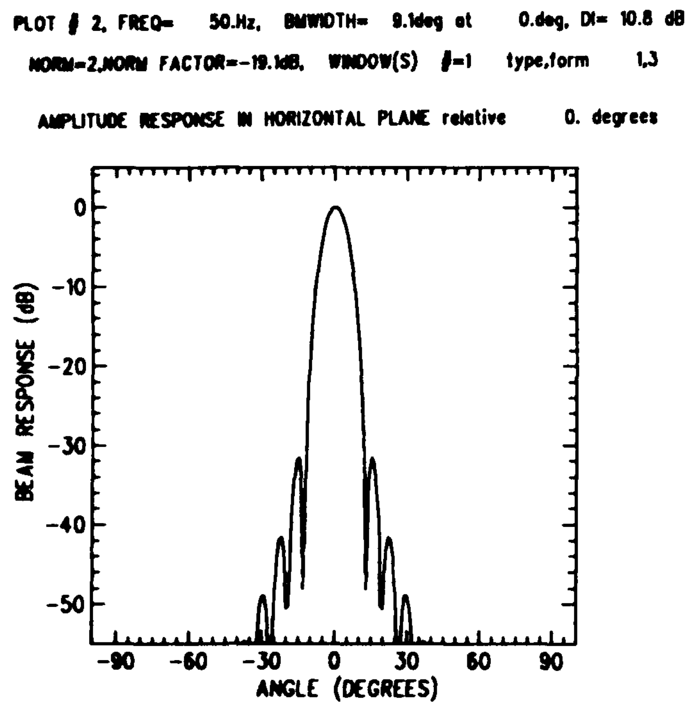


Figure 9. Amplitude response of the broadside beam of a 17 element horizontal line array at 50 Hz. The beam is sampled in the horizontal plane. A Hann window has been applied before beamforming.

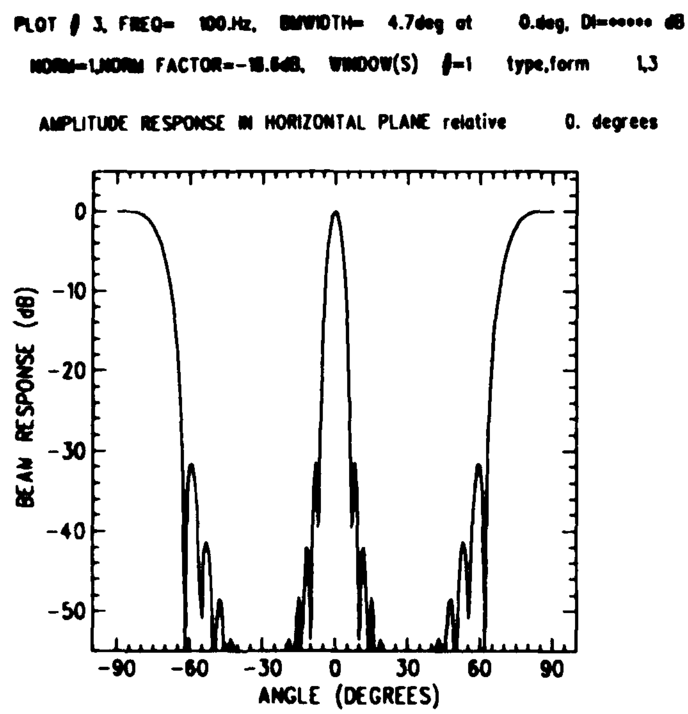


Figure 10. Amplitude response in the horizontal plane of a broadside beam of the 17 element horizontal line at 100 Hz. A Hann window has been applied before beamforming.

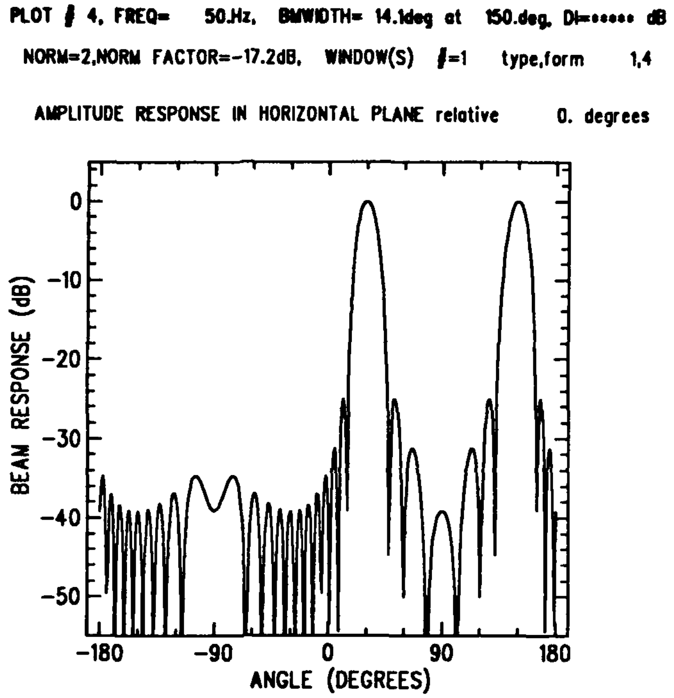


Figure 11. Amplitude response in the horizontal plane at 50 Hz, of a horizontal line array steered to a bearing 30° forward of broadside. A Hamming window has been applied before beamforming.

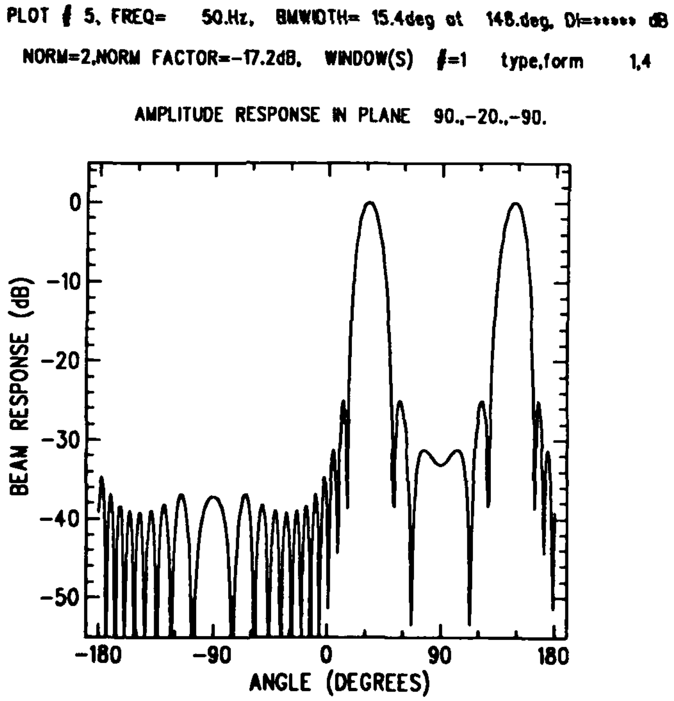


Figure 12. Amplitude response for the same beam as shown in Figure 10 sampled in a plane 20° out of the horizontal. This gives a response that is equivalent to the response in the horizontal plane when the array is tilted downward 20°.

4. SUMMARY

This program provides a flexible tool for calculating and examining the amplitude and phase response of arrays. General three-dimensional arrays containing directional elements can be handled. Both the effects of array distortion and element phase and amplitude imbalance can be investigated using BMPAT. A great deal of flexibility is built into the beamforming process, although beamforming is constrained to be a linear operation. Beamwidths are calculated and the beam directivity index can optionally be calculated. The output is directed to SAPLOT formatted files from which the data can either be plotted or accessed for other purposes.

REFERENCES

1. P. R. Staal, DREA Surveillance Acoustics Plotting Package (SAPLOT), private communication

2. F.J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proc. IEEE, Vol. 66, #1, pp51-83, Jan 78.

3. B.A. Trenholm and C.A. Young, "Application of Taylor Shading to a Line Array of Equally Spaced Elements", DREA Research Note AM/77/4, July 1977, Informal Communication.

APPENDIX A
DIALOGUE, INPUT AND OUTPUT FILES FOR CIRCULAR ARRAY EXAMPLE

A.1 Input file containing array geometry

The contents of the input file CARD1.DAT used for inputting the array geometry was as follows: (see section 2.4.3 on page 4)

```
Geometry of a circular array of 3 cardioid & 3 dipole elements
4
Y
6
10.,0.,0.,3,0.,0.
10.,60.,0.,2,60.,0.
10.,120.,0.,3,120.,0.
10.,180.,0.,2,180.,0.
10.,240.,0.,3,240.,0.
10.,300.,0.,2,300.,0.
1
```

A.2 Input dialogue

Following is the dialogue for the run in which the beam patterns for the circular array were generated. The computer output is shown in small font, the keyboard input is in emboldened normal sized font and comments are in standard font.

```
RUN BMPAT
Program BMPAT for calculation of array amplitude and phase response

*****Menu for controlling program operation*****
  1. Select keyboard for input
  2. Select a file for data input
  3. Enter frequency and sound speed
  4. Input array geometry and element types
  5. Enter ideal (undistorted) array geometry
  6. Enter a table defining element response
  7. Enter element amplitude and phase imbalances
  8. Enter a steering vector
  9. Define amplitude windows
  10. Calculate phase to a plane steering vector
  11. Calculate and output beam amplitude, phase  response and/or directivity
  12. Open output files
  13. Exit
Enter either menu # or M to retype menu? <M>?? 2
Enter file name containing input data: CARD1
This file is described as follows
Geometry of a circular array of 3 cardioid & 3 dipole elements
Is this the correct file? <Y>?? Y
```

The following text (between the ---------- marks) is output as the file input is read. In some functional steps the output of some of the keyboard prompts is not disabled while data is being input from a file; this allows some monitoring of program operation when the active input is a file.

```
-----------------------------------------------------------------
Type of coord system in which array to be defined
     cartesian(C), cylindrical(Y) or polar(P)? <C>??
Enter the number of array elements:
Enter X,Y,Z,ELEMENT TYPE,element-axis AZIMUTHAL,POLAR angles for each element
 element 1?
 element 2?
 element 3?
 element 4?
 element 5?
 element 6?
-----------------------------------------------------------------
Enter either menu # or M to retype menu? <M>?? 3
Enter the frequency? 100
Enter the sound speed? <1500 m/sec>?? 4800
Enter either menu # or M to retype menu? <M>?? 10
Phase to a plane beam steering selected
Enter the beam steering direction in the array coord system
                                  - azimuthal,polar angles (deg): 0,0
Use actual(A) or ideal(I) element positions for steering? <A>?? A
Enter either menu # or M to retype menu? <M>?? 11
Is beam to be sampled in the horizontal(H)
                              vertical(V)
                              or other(O) plane? <H>?? H
Enter azimuthal angle to which to reference beam response? 0
Enter angles at which beam is to be sampled: min,max,increment? 0,360,1
Enter the type of normalization to employ where:
          1. sets the peak response to 0 dB,
          2. normalizes relative the beam response in the steering direction,
          3. normalizes relative the 3-dimensional peak (NOTE this is slow)
                                                       1,2 or 3? <1>?? 1
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? Y
Enter angular step size for peak search/directivity calculation? <0.5>?? 2
Enter the minimum beam power to plot: < -10>?? -40
```

At this point the first plot is complete.

```
Enter either menu # or M to retype menu? <M>?? 11
Is beam to be sampled in the horizontal(H)
                              vertical(V)
                              or other(O) plane? <H>?? V
Azimuthal angle defining orientation of vertical sample plane? <0>?? 180
Enter angles at which beam is to be sampled: min,max,increment?? -180,180,1
Enter the type of normalization to employ where:
          1. sets the peak response to 0 dB,
          2. normalizes relative the beam response in the steering direction,
          3. normalizes relative the 3-dimensional peak (NOTE this is slow)
                                                       1,2 or 3? <1>?? 1
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>??N
Enter the minimum beam power to plot: < -16>?? -40
```

At this point the second plot is complete.

```
Enter either menu # or M to retype menu?  <M>?? 3
Enter the frequency? 1000
Enter the sound speed? <1500 m/sec>??  4800
Enter either menu # or M to retype menu?  <M>?? 10
Phase to a plane beam steering selected
Enter the beam steering direction in the array coord system - azimuthal,polar angles (deg):  0,0
Use actual(A) or ideal(I) element positions for steering? <A>??  A
Enter either menu # or M to retype menu?  <M>?? 11
Is beam to be sampled in the horizontal(H)
                              vertical(V)
                              or other(O) plane? <H>?? H
Enter azimuthal angle to which to reference beam response? 0
Enter angles at which beam is to be sampled:  min,max,increment?? -180,180,1
Enter the type of normalization to employ where:
        1. sets the peak response to 0 dB,
        2. normalizes relative the beam response in the steering direction,
        3. normalizes relative the 3-dimensional peak (NOTE this is slow)   1,2 or 3? <1>?? 1
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? N
Enter the minimum beam power to plot:  < -40>??  -40
Enter either menu # or M to retype menu?  <M>?? 13
FORTRAN STOP
```

Note in the above dialogue that after the frequency was changed by calling step 3 the steering vector was recalculated via step 10. If step 10 was not called after the frequency change the 100 Hz steering vector would have been used and the result would have been the response at 1000 Hz for 100 Hz steering vectors.

A.3 Output SAPLOT files

The output files resulting from this run were the default files BMPAT.AMP and BMPAT.PHS. The contents of the file BMPAT.AMP are listed below (this listing has been edited to reduce its length). Comments are inserted in italics.

```
ORSET  3.50  1.50                          ORSET, AXSET & THSET  set up the plot area
AXSET  5.50  5.50                          they are only output once to the file
THSET  90.00
CHSET  100.00 100.00                       CHSET scales the characters
SETSPD  1  100
RANGE 1 -190.00 190.00                     setup of axes
RANGE 3 -190.00 190.00                     "
SSET  1 -180.00  90.00   3                 "
SSET  3 -180.00  90.00   3                 "
 NSET  3  0                                "
RANGE 2  -45.00   5.00                     "
SSET  2  -40.00  10.00   5                 "
RANGE 4  -45.00   5.00                     "
SSET  4  -40.00  10.00   5                 "
 NSET  4  0                                "
 LABEL 3                                   information headers
!SCA0.6;AMPLITUDE RESPONSE IN HORIZONTAL PLANE relative    0. degrees   "
 LABEL 7                                   "
WINDOW(S)  #=0                             "
 LABEL 11                                  "
```

```
PLOT # 1, FREQ= 100.Hz, NORM=2, BMWIDTH=102.3deg at -9.deg, DI= -4.2 dB!SCA1.667;      "
 LABEL 1                                                                   axes labels
 ANGLE (DEGREES)                                                           "
 LABEL 2                                                                   "
 BEAM RESPONSE (dB)                                                        "
 CURVE                                                                     data curve
 -180.0000    -3.842369
 -179.0000    -3.845440
 -178.0000    -3.854662
    -             -
    -             -
    -             -
  178.0000    -3.854662
  179.0000    -3.845440
  180.0000    -3.842369
 NEWPAGE
 PLOT
CHSET 100.00 100.00                                                        start of second plot
SETSPD 1 100
RANGE 1 -190.00 190.00
RANGE 3 -190.00 190.00
SSET 1 -180.00 90.00 3
SSET 3 -180.00 90.00 3
 NSET 3 0
RANGE 2 -45.00 5.00
SSET 2 -40.00 10.00 5
RANGE 4 -45.00 5.00
SSET 4 -40.00 10.00 5
 NSET 4 0
LABEL 3
!SCA0.6;AMPLITUDE RESPONSE IN VERTICAL PLANE BEARING 0. degrees
 LABEL 7
WINDOW(S) #=0
 LABEL 11
PLOT # 2, FREQ= 100.Hz, NORM=1, BMWIDTH=164.3deg at -33.deg, DI=***** dB!SCA1.667;
 LABEL 1
 ANGLE (DEGREES)
 LABEL 2
 BEAM RESPONSE (dB)
 CURVE
 -180.0000    -3.842369
 -179.0000    -3.843784
    -             -
    -             -
    -             -
  179.0000    -3.843784
  180.0000    -3.842369
 NEWPAGE
 PLOT
CHSET 100.00 100.00                                                        start of third plot
       -
       -
       -
```

APPENDIX B
INPUT FILE AND DIALOGUE FOR 17 ELEMENT LINE ARRAY EXAMPLE

B.1 Input file containing array geometry

The contents of the input file LINE1.DAT containing the array geometry was as follows:

```
This file defines a 17 element ln array along y-axis cut for 50 Hz, C=1500 m/s
4
C
17
0,-120,0,0,0,0
0,-105,0,0,0,0
0,-90,0,0,0,0
0,-75,0,0,0,0
0,-60,0,0,0,0
0,-45,0,0,0,0
0,-30,0,0,0,0
0,-15,0,0,0,0
0,0,0,0,0,0
0,15,0,0,0,0
0,30,0,0,0,0
0,45,0,0,0,0
0,60,0,0,0,0
0,75,0,0,0,0
0,90,0,0,0,0
0,105,0,0,0,0
0,120,0,0,0,0
1
```

B.2 Input dialogue

Following is the dialogue for the run in which the beam patterns for the line array were generated. The computer output is shown in small font, the keyboard input in emboldened normal sized font and comments are in standard font.

R BMPAT

Program BMPAT for calculation of array amplitude and phase response

```
*****Menu for controlling program operation*****
 1. Select keyboard for input
 2. Select a file for data input
 3. Enter frequency and sound speed
 4. Input array geometry and element types
 5. Enter ideal (undistorted) array geometry
 6. Enter a table defining element response
 7. Enter element amplitude and phase imbalances
 8. Enter a steering vector
 9. Define amplitude windows
10. Calculate phase to a plane steering vector
11. Calculate and output beam amplitude, phase  response and/or directivity
12. Open output files
13. Exit
```

Enter either menu # or M to retype menu? <M>?? **2**
Enter file name containing input data: **LINE1**
This file is described as follows
This file defines a 17 element ln array along y-axis cut for 50 Hz, C=1500 m/s
Is this the correct file? <Y>?? **Y**

The following text (between the ------- marks) is output as the file input is read. In some functional steps the output of some of the keyboard prompts is not disabled while data is being input from a file; this allows some monitoring of program operation when the active input is a file.

```
----------------------------------------------------------------
Type of coord system in which array to be defined
                                        cartesian(C), cylindrical(Y) or polar(P)? <C>??
Enter the number of array elements:
Enter X,Y,Z,ELEMENT TYPE,element-axis AZIMUTHAL,POLAR angles for each element
 element  1?
 element  2?
 element  3?
 element  4?
 element  5?
 element  6?
 element  7?
 element  8?
 element  9?
 element 10?
 element 11?
 element 12?
 element 13?
 element 14?
 element 15?
 element 16?
 element 17?
----------------------------------------------------------------
```

Enter either menu # or M to retype menu? <M>?? **3**
Enter the frequency? **50**
Enter the sound speed? <1500 m/sec>?? **1500**
Enter either menu # or M to retype menu? <M>?? **10**
Phase to a plane beam steering selected
Enter the beam steering direction in the array coord system
- azimuthal,polar angles (deg)? **0,0**
Use actual(A) or ideal(I) element positions for steering? <A>?? **A**
Enter either menu # or M to retype menu? <M>?? **12**
Enter name for output files (include no extension)? **LINE**
Enter either menu # or M to retype menu? <M>?? **11**
Is beam to be sampled in the horizontal(H)
vertical(V)
or other(O) plane? <H>?? **H**
Enter azimuthal angle to which to reference beam response? **0**
Enter angles at which beam is to be sampled: min,max,increment?? **-90,90,1**

Enter the type of normalization to employ where:
1. sets the peak response to 0 dB,
2. normalizes relative to the beam response in the steering direction,
3. normalizes relative to the 3-dimensional peak (NOTE this is slow)
1,2 or 3? <1>?? **2**

Use response of actual(A) or ideal(I) array for normalization? <A>?? **A**

Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? **Y**

Enter angular step size for peak search/ directivity calculation? <0.5>?? **1**

Enter the minimum beam power to plot: <-100>?? **-50**

This completes the input for the first plot.

Enter either menu # or M to retype menu? <M>?? **9**
Definition of window amplitude functions
Enter E to input a window table
W to define windows
T to enable/disable windows? <W>?? **W**
Define the amplitude window to use in beamforming
Enter the # of independent windows to apply (max 2)? **1**
* Enter parameters defining window 1 **
Select the window type as follows:
enter 1 to select angle indep horizontal window
2 to select angle depen horizontal window
3 to select angle indep vertical window
4 to select angle depen vertical window
5 to select angle indep 2-d window
6 to select angle depen 2-d window ?? **1**
Enter the azimuthal angle defining the window axis? **90**

Select the window form as follows:
enter 0 to select a rectangular window
1 to define window from table 1
2 to define window from table 2
3 to select a Hann window
4 to select a Hamming window
5 to select a raised cosine window
6 to select a raised cos**2 window
7 to select a Gaussian window
8 to select a Taylor window ?? **3**
Compensate window weights for unequal spacing of elements over the window aperture? (NOTE that this option should be used with caution!) Y/N? <N>?? **N**
Enter either menu # or M to retype menu? <M>?? **11**
Is beam to be sampled in the horizontal(H)
vertical(V)
or other(O) plane? <H>?? **H**
Enter azimuthal angle to which to reference beam response? **0**
Enter angles at which beam is to be sampled: min,max,increment?? **-90,90,1**
Enter the type of normalization to employ where:
1. sets the peak response to 0 dB,
2. normalizes relative to the beam response in the steering direction,
3. normalizes relative to the 3-dimensional peak (NOTE this is slow)
1,2 or 3? <1>?? **2**

Use response of actual(A) or ideal(I) array for normalization? <A>?? **A**
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? **Y**
Enter angular step size for peak search/ directivity calculation? <0.5>?? **2**
Enter the minimum beam power to plot: <-100>?? **-50**

This completes the input for the second plot.

Enter either menu # or M to retype menu? <M>?? 3
Enter the frequency? 100
Enter the sound speed? <1500 m/sec>?? 1500
Enter either menu # or M to retype menu? <M>?? 10
Phase to a plane beam steering selected
Enter the beam steering direction in the array coord system
- azimuthal,polar angles (deg): 0,0
Use actual(A) or ideal(I) element positions for steering? <A>?? A
Enter either menu # or M to retype menu? <M>?? **11**
Is beam to be sampled in the horizontal(H)
vertical(V)
or other(O) plane? <H>?? **H**
Enter azimuthal angle to which to reference beam response? **0**
Enter angles at which beam is to be sampled: min,max,increment?? **-90,90,1**
Enter the type of normalization to employ where:
1. sets the peak response to 0 dB,
2. normalizes relative to the beam response in the steering direction,
3. normalizes relative to the 3-dimensional peak (NOTE this is slow)
1,2 or 3? <1>?? **1**
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? **N**
Enter the minimum beam power to plot: <-100>?? **-50**

This completes the input for the third plot.

Enter either menu # or M to retype menu? <M>?? **3**
Enter the frequency? **50**
Enter the sound speed? <1500 m/sec>?? **1500**
Enter either menu # or M to retype menu? <M>?? **9**
Definition of window amplitude functions
Enter E to input a window table
W to define windows
T to enable/disable windows? <W>?? **W**
Define the amplitude window to use in beamforming
Enter the # of independent windows to apply (max 2)? **1**
* Enter parameters defining window 1 **
Select the window type as follows:
enter 1 to select angle indep horizontal window
2 to select angle depen horizontal window
3 to select angle indep vertical window
4 to select angle depen vertical window
5 to select angle indep 2-d window
6 to select angle depen 2-d window ?? **1**
Enter the azimuthal angle defining the window axis? **90**

Select the window form as follows: enter 0 to select a rectangular window
1 to define window from table 1
2 to define window from table 2
3 to select a Hann window
4 to select a Hamming window
5 to select a raised cosine window
6 to select a raised cos**2 window
7 to select a Gaussian window
8 to select a Taylor window ?? **4**
Compensate window weights for unequal spacing of elements over the window
aperture? (NOTE that this option should be used with caution!) Y/N? <N>?? **N**
Enter either menu # or M to retype menu? <M>?? **10**
Phase to a plane beam steering selected. Enter the beam steering direction in the array coord system
- azimuthal,polar angles (deg)? **30,0**
Use actual(A) or ideal(I) element positions for steering? <A>?? **A**
Enter either menu # or M to retype menu? <M>?? **11**
Is beam to be sampled in the horizontal(H)
vertical(V)
or other(O) plane? <H>?? **H**
Enter azimuthal angle to which to reference beam response? **0**
Enter angles at which beam is to be sampled: min,max,increment?? **-180,180,1**
Enter the type of normalization to employ where:
1. sets the peak response to 0 dB,
2. normalizes relative the beam response in the steering direction,
3. normalizes relative the 3-dimensional peak (NOTE this is slow)
1,2 or 3? <1>?? **2**
Use response of actual(A) or ideal(I) array for normalization? <A>?? **A**
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? **N**
Enter the minimum beam power to plot: <-100>?? **-50**

This completes the input for the fourth plot.

Enter either menu # or M to retype menu? <M>?? **11**
Is beam to be sampled in the horizontal(H)
vertical(V)
or other(O) plane? <H>?? **O**
Enter the z,y' rotations defining the sample plane? **90,-20**
With these z,y' rotations the z" rotation required to put the beam x-axis
in the north plane is -90.0 degrees
Enter z" rotation (orients x-axis of the beam system in the x-y plane)? **-90**
Enter angles at which beam is to be sampled: min,max,increment?? **-180,180,1**
Enter the type of normalization to employ where:
1. sets the peak response to 0 dB,
2. normalizes relative to the beam response in the steering direction,
3. normalizes relative to the 3-dimensional peak (NOTE this is slow)
1,2 or 3? <1>?? **2**
Use response of actual(A) or ideal(I) array for normalization? <A>?? **A**
Is directivity index to be calculated - THIS IS TIME CONSUMING - (Y/N)? <N>?? **N**
Enter the minimum beam power to plot: <-100>?? **-50**
Enter either menu # or M to retype menu? <M>?? **13**
FORTRAN STOP

This completes the input for the fifth and final plot.

APPENDIX C

PROGRAM LISTING

```
      PROGRAM BMPAT
*****************************************************************
*****************************************************************
*   Program for calculating and plotting the amplitude and phase response
*   of general 3-dimensional sonar arrays.
*****************************************************************
*****************************************************************
*
*   A. Collier 23 Feb/87     original 18 Aug/86
*
*********************************************************************
*   The descriptive header, which duplicates much of the information in the
*        text of this report, is not included in this listing.
*********************************************************************
*
*  REQUIRED SUBROUTINES and FUNCTIONS
*
*       OPNFIL            prompts for a file name and opens a formatted file
*       VECX3(V1,V2)      calculates the cross product of 3-D vectors V1 and V2
*                                    V1 = V1 X V2
*       TBINTP            for interpolation from tabulated values
*
*       ROTXFM, ZYZXFM and ROTATE These subroutines are used to make the
*                                       rotational transformations between the
*                                       array and beam co-ordinate systems.
*
*       Functions JIZERO and SUM  are functions used in calculating the modified Bessel
*               Function of order zero (IO)
*
*   COMPILATION AND LINK COMMANDS
*       FOR   [COLLIER.ARRAY]BMPAT,ROTXFM
*       LINK   BMPAT,ROTXFM
*
******************************************************************
******************************************************************

      PARAMETER                   MXONE=100,                  'max #elements
     ^                            MXWIN=2,            'max # windows
     ^                            MXNTB=2,            'max # tables
     ^                            MXTBEN=100,                 'max entries table
     ^                            MXNB=720,                   'max # smps of bm pat
     ^                            LO1=30,                     LUN for o/p of bm pw
     ^                            LO2=31,                     LUN for o/p of bm phs
     ^                            PR1=3.5,            'plot origin " over
     ^                            PR2=1.5,                    plot origin " up
     ^                            PR3=5.5,            'plot frame height "
     ^                            PR4=5.5,            'plot frame width "
     ^                            PR5=90                      plot rotation angle

      LOGICAL   ERR,TTYINP,TMENU,USEACTUAL,LOGT,PASS1,
     ^          STEP3/.FALSE./,STEP4/.FALSE./,STEP5/.FALSE./,
     ^          STEP6/.FALSE./,STEP8/.FALSE./,STEP10/.FALSE./,
     ^          STEP11/.FALSE./,STEP12/.FALSE./,
     ^          WFLAG/.FALSE./,EWFLAG/.FALSE./,IDEAL,CALDIR,PEAK
      CHARACTER         COORT*1,CHR*2
      BYTE      NAME(80),TEXT(80)
      INTEGER   IETYP(MXONE),IWTYP(MXWIN),IWFRM(MXWIN),NEWTBL(MXNTB),
     ^          IEMUL(MXONE),IPKPOS(20)
      REAL*4    X(MXONE),Y(MXONE),Z(MXONE),XI(MXONE),YI(MXONE),ZI(MXONE),
     ^          EEPOL(MXONE),EEAZ(MXONE),     AE(MXONE),PE(MXONE),
     ^          WV(3),PX(MXONE),PY(MXONE),WTHET(MXWIN),WPHI(MXWIN),
     ^          WPAR(MXWIN),WTBL(2*MXTBEN,MXNTB),
     ^          XB(MXONE),YB(MXONE),ZB(MXONE),ELETBL(3*MXTBEN),
     ^          ANGO(MXNB),BMP(MXNB),PHS(MXNB),WX(MXONE),WY(MXONE),
     ^          WZ(MXONE),AS(MXONE),AX(MXONE),BX(MXONE),AY(MXONE),BY(MXONE),
     ^          AZ(MXONE),BZ(MXONE),WWT(MXONE)/MXONE*1./,XM(3),YM(3)
      COMPLEX   TC,CPW,CPW1,CPW2,SW(MXONE),CE
      DATA      AE/MXONE*1./,      PE/MXONE*0./,       'init ampl & phs factors
     ^          PI/3.1415926535/            'pi
      EQUIVALENCE          (NAME,TEXT)

      NPLOT = 0                                        'initialization of plot #
      DTR = PI/180.                                    'degrees to radians
      LDT = 5                                          'initially input set to TTY
      R1=PR1
      R2=PR2
      R3=PR3
      R4=PR4
      R5=PR5

      TYPE *,'Program BMPAT for calculation of array amplitude and phase
     ^ response'
      TYPE *,' '
      TMENU = .TRUE.                                   'initialization for typing menu
*     Type Menu and/or select program function by selecting from menu
20    IF(LDT.EQ.5) THEN
        IF(TMENU) THEN                          'type menu if operating fm keybd
          TMENU = .FALSE.
          TYPE *,'*****Menu for controlling program operation*****'
          TYPE *,'    1. Select keyboard for input'
          TYPE *,'    2. Select a file for data input'
          TYPE *,'    3. Enter frequency and sound speed'
```

```
            TYPE *,'    4. Enter array geometry and element types'
            TYPE *,'    5. Enter ideal (undistorted) array geometry'
            TYPE *,'    6. Enter a table defining element response'
            TYPE *,'    7. Enter element amplitude and phase imbalances'
            TYPE *,'    8. Enter a steering vector'
            TYPE *,'    9. Define amplitude windows'
            TYPE *,'   10. Calculate phase to a plane steering vector'
            TYPE *,'   11. Calculate and output beam amplitude, phase
     * response and/or directivity'
            TYPE *,'   12. Open output files'
            TYPE *,'   13. Exit'
            TYPE *,' '
          END IF
          TYPE 2210
2210      FORMAT ('$Enter either menu # or M to retype menu?  <M>?? ')
        END IF
        READ (LDT,2215) NC,CHR
2215    FORMAT (Q,A2)
        IF(NC.EQ.0.OR.CHR(1:1).EQ.'M'.OR.CHR(1:1).EQ.'m') THEN
          TMENU=.TRUE.
          GOTO 20
        ELSE
          IF(NC.EQ.1) THEN
           DECODE (1,2220,CHR(1:1),ERR=20) MENU
2220       FORMAT (I1)
          ELSE IF(NC.EQ.2) THEN
           DECODE (2,2230,CHR,ERR=20) MENU
2230       FORMAT (I2)
          ELSE
           GOTO 20
          END IF
          IF(MENU.LT.1.OR.MENU.GT.13) GOTO 20
          GOTO (1,2,3,4,5,6,7,8,9,10,11,12,13) MENU
        END IF
**************************************************************
*       STEP 1  Select input to be from keyboard
 1      IF(LDT.NE.5) CLOSE (UNIT=LDT,ERR=100)
100     LDT = 5
        GOTO 20
**************************************************************
*       STEP 2  Select a file for data input
 2      TTYINP = .TRUE.
        IF(LDT.NE.5) THEN
          CLOSE (UNIT=LDT,ERR=200)
          TTYINP = .FALSE.
        END IF
        LPDT = LDT
200     IF (LPDT.EQ.5.OR.LPDT.EQ.21) LDT = 20
        IF (LPDT.EQ.20) LDT = 21
210     CALL OPNFIL(LDT,'Enter file name containing input data:  !',
     .                                               LPDT,ERR)
        IF(ERR) THEN
          IF(TTYINP) THEN                  !last input was from TTY
            TYPE *,'Error in opening input data file.  TRY AGAIN'
            GOTO 200
          ELSE                                     !last input was from a file
            STOP 'Error in opening input data file'
          END IF
        END IF

        READ (LDT,2000) NC,TEXT
2000    FORMAT (Q,80A1)
        TYPE *,'This file is described as follows'
        TYPE 2010,(TEXT(J),J=1,NC)
2010    FORMAT (' ',80A1)
        TYPE 2020
2020    FORMAT ('$Is this the correct file?  <Y>?? ')
        ACCEPT 2215,NC,CHR
        IF(CHR(1:1).EQ.'N'.OR.CHR(1:1).EQ.'n') THEN
          CLOSE (UNIT=LDT)
          GOTO 210
        END IF
        GOTO 20
*******************************************************************

*******************************************************************
*       STEP 3 Enter the frequency, sound speed
 3      STEP3=.TRUE.
        TYPE 3000
3000    FORMAT ('$Enter the frequency? ')
        READ (LDT,*) FREQ
        TYPE 3010
3010    FORMAT ('$Enter the sound speed? <1500 m/sec>??  ')
        READ (LDT,3020) NC,C
3020    FORMAT (Q,F10.0)
        IF(NC.EQ.0) C=1500.
        WVNO = 2.*PI*FREQ/C                          !wavenumber
        GOTO 20
*******************************************************************
```

```
***********************************************************************
*       STEP 4   Input of the element locations, types & orientations

 4      STEP4=.TRUE.
        TYPE *,'Type of coord system in which array to be defined'
*       Cartesian coord, X,Y,Z   Cylindrical coords, X=R,Y=PHI,Z=Z
*                                    Polar coords X=R,Y=PHI,Z=THETA
*               Where R is radius, PHI is azimuthal angle,
*               THETA is polar angle (0 at horizontal and positive above hor)
*       The units are established by the speed of sound input in step 3

        TYPE 4000
 4000   FORMAT ('$         cartesian(C), cylindrical(Y) or
     ^ polar(P)?  <C>??  ')
        READ (LDT,4010) NC,COORT  !type of coords for input of ele pstns
 4010   FORMAT (Q,A1)

        IF(NC.EQ.0) COORT='C'
        TYPE 4020
 4020   FORMAT ('$Enter the number of array elements:  ')
        READ (LDT,*) NEL                   !# of elements
        TYPE *,'Enter X,Y,Z,ELEMENT TYPE,element-axis
     ^ AZIMUTHAL,POLAR angles for each element'
*       Read ele coords and convert element positions to Cartesian coords
        DO 400 IE=1,NEL
          TYPE 4030,IE
 4030     FORMAT ('$  element ',I2,'?  ')
          READ (LDT,*) X(IE),Y(IE),Z(IE),IETYP(IE),EEAZ(IE),EEPOL(IE)
          IF(COORT.NE.'C'.AND.COORT.NE.'c') THEN
            R=X(IE)
            IF(COORT.EQ.'Y'.OR.COORT.EQ.'y') THEN
              PHI=Y(IE)
              X(IE) = R*COSD(PHI)
              Y(IE) = R*SIND(PHI)
            ELSE IF (COORT.EQ.'P'.OR.COORT.EQ.'p') THEN
              PHI = Y(IE)
              THETA= Z(IE)
              X(IE) = R*COSD(THETA)*COSD(PHI)
              Y(IE) = R*COSD(THETA)*SIND(PHI)
              Z(IE) = R*SIND(THETA)
            ELSE
              TYPE *,'The character defining the coord system is
     ^ incorrect'
              STOP
            END IF
          END IF
 400    CONTINUE
        GOTO 20
***********************************************************************

***********************************************************************
*       STEP 5  Enter ideal (undistorted) element positions for steering vector
*               calculations.

 5      STEP5=.TRUE.
        TYPE *,'Type of coordinate system in which assumed positions are
     ^ to be input'
        TYPE 4000
        READ (LDT,4010) NC,COORT  !type of coords for input of ele pstns
        IF (NC.EQ.0) COORT = 'C'
        TYPE *,'Enter ideal element positions for steering
     ^ vector calculations'
*       Input and convert element positions to Cartesian coords
        DO 500 IE=1,NEL
          TYPE 5000, IE
 5000     FORMAT ('$  ele ',I2,' coords? ')
          READ (LDT,*) XI(IE),YI(IE),ZI(IE)
          IF(COORT.NE.'C'.AND.COORT.NE.'c') THEN
            R=XI(IE)
            IF(COORT.EQ.'Y'.OR.COORT.EQ.'y') THEN
              PHI=YI(IE)
              XI(IE) = R*COSD(PHI)
              YI(IE) = R*SIND(PHI)
            ELSE IF (COORT.EQ.'P'.OR.COORT.EQ.'p') THEN
              PHI = ZI(IE)
              THETA=ZI(IE)
              XI(IE) = R*COSD(THETA)*COSD(PHI)
              YI(IE) = R*COSD(THETA)*SIND(PHI)
              ZI(IE) = R*SIND(THETA)
            ELSE
              TYPE *,'The character defining the coord system is
     ^ incorrect'
              IF(LDT.EQ.5) GOTO 5
              STOP
            END IF
          END IF
 500    CONTINUE
        GOTO 20
***********************************************************************

***********************************************************************
*       STEP 6 Input of element response table.
```

```
*       The element response is assumed to axially symmetric.  The response
*         must be entered for monotonically increasing independent variables.
*         The independent variable is the angle relative the element axis in
*         degrees which normally goes from 0 to 180 degrees.  Phase angles are
*         to be in the range +,- 360 deg.
 6      STEP6=.TRUE.
        TYPE *,'Enter the element response table'
        TYPE *,'  angles increasing monotonically from 0 to 180 deg'
        TYPE *,'  Terminate entries by entering -1'
        TYPE *,'     Angle(deg), amp response(dB), phase response(deg)'
        DO 600 IT=1,MXTBEN
          IX=(IT-1)*3+1
          READ (LDT,6000) NC,ELETBL(IX),ELETBL(IX+1),ELETBL(IX+2)
 6000     FORMAT (Q,3F10.0)
          IF(NC.EQ.2.AND.ELETBL(IX).EQ.-1.) THEN
            NELTB = IT-1
            GOTO 20
          END IF
 600    CONTINUE
        TYPE *,'***Number of entries in table exceeds array dimension'
        IF (LDT.NE.5) STOP
        GOTO 20
************************************************************************

************************************************************************
*       STEP 7 Input of element amplitude and phase imbalance

 7      TYPE *,'Element amplitude and phase imbalance'
        TYPE 7000
 7000   FORMAT ('Enter: Z to zero errors,D for deterministic errors,
     ^ R for random errors? <R>?? ')
        READ (LDT,2215) NC,CHR
        IF(CHR(1:1).EQ.'D'.OR.CHR(1:1).EQ.'d') THEN
          TYPE 7020,NEL
 7020     FORMAT (' Enter the amplitude factor, phase error (deg)
     ^ for the ',I3,' elements')
          READ (LDT,*) (AE(IE),PE(IE),IE=1,NEL)
        ELSE IF (CHR(1:1).EQ.'Z'.OR.CHR(1:1).EQ.'z') THEN
          DO 700 IE = 1,NEL
            PE(IE) = 0.
            AE(IE) = 1.
 700      CONTINUE
          TYPE 7030
 7030     FORMAT ('$Enter the seed for random # generation:  ')
          READ (LDT,*) ISEED
          TYPE 7040
 7040     FORMAT ('$Enter the max amplitude deviation (dB), max phase
     ^ deviation (deg):  ')
          READ (LDT,*) AMPER,PHER
          AERMX = 10.**(AMPER/20.)
          AERMN = 1.-AERMX
          ARNG = AERMX-AERMN
          AMEAN = (AERMX+AERMN)/2
          DO 710 IE=1,NEL
            AE(IE) = ARNG* RAN(ISEED) - (AERMX-AERMN)/2
            PE(IE) = PHER* (RAN(ISEED)-.5)
 710      CONTINUE
        END IF
        GOTO 20
************************************************************************
************************************************************************
*       STEP 8 Enter a complex steering vector

 8      STEP8=.TRUE.
        STEP10 = .FALSE.  !steering vectors not calculated
        TYPE 8000,NEL
 8000   FORMAT ('$Enter a complex steering vector of order ',I2)
        DO 800 IE=1,NEL
          TYPE 8010,IE
 8010     FORMAT ('$  complex wt for element ',I2,'?  ')
          READ (LDT,*) RWT,AIWT
          SW(IE) = CMPLX(RWT,AIWT)
 800    CONTINUE
        GOTO 20
************************************************************************

************************************************************************
*       STEP 9  Define amplitude windows
*       It is assumed that the windows are all symmetric and thus tables
*               of window functions need only go from normalized apertures
*               of 0 to 1

 9      TYPE *, 'Definition of window amplitude functions'
        TYPE *,'Enter     E to input a window table'
        TYPE *,' W to define windows'
        TYPE 9000
 9000   FORMAT ('$         T to enable/disable windows?       <W>??  ')
        READ (LDT,2215) NC,CHR

        IF(CHR(1:1).EQ.'E'.OR.CHR(1:1).EQ.'e') THEN         !enter tables
          TYPE 9010,MXNTB
 9010     FORMAT ('$Enter the number to be used to identify table
```

```
     ^ (max=',I2,')? ')
          READ (LDT,*) NTBL
          TYPE *,'Enter pairs of coords defining window:
     ^  abcissa,ordinate'
          TYPE *,'       where abcissa=normalized aperture (0 to 1) and
     ^ ordinate=window amplitude'
          TYPE *,' -1 terminates entry'
          NEWTBL(NTBL)=0
          DO 900 IT=1,MXTBEN
            TYPE 9020
9020        FORMAT ('$  ??  ')
            READ (LDT,9025) NC,ABSI,ORD
9025        FORMAT (Q,2F10.0)
            IF(ABSI.EQ.-1.) GOTO 20
            NEWTBL(NTBL)=NEWTBL(NTBL)+1
            WTBL(IT*2-1,NTBL) = ABSI
            WTBL(IT*2,NTBL)    = ORD
900       CONTINUE
          TYPE *,'Window amplitude table has been filled before table
     ^ entry terminated'
          IF(LDT.NE.5) STOP
          GOTO 20
        ELSE IF (CHR(1:1).EQ.'T'.OR.CHR(1:1).EQ.'t') THEN   !turn window on/off
          IF (EWFLAG) WFLAG = .NOT.WFLAG  !can only toggle when a window defined
        ELSE
          TYPE *,'Define the amplitude window to use in beamforming'
          TYPE 9040
9040      FORMAT ('$Enter the # of independent windows to apply
     ^ (max 2)? ')
          READ(LDT,*) NWIND
          DO 910 IW=1,NWIND
            TYPE 9050,IW
9050        FORMAT ('** Enter parameters defining window ',I1,' **')
            TYPE *,'Select the window type as follows:'
            TYPE *,'      enter 1 to select angle indep horizontal window'
            TYPE *,'            2 to select angle depen horizontal window'
            TYPE *,'            3 to select angle indep vertical window'
            TYPE *,'            4 to select angle depen vertical window'
            TYPE *,'            5 to select angle indep 2-d window'
            TYPE 9060
9060        FORMAT ('$          6 to select angle depen 2-d window
     ^       ?? ')
            READ (LDT,*) IWTYP(IW)
            IF(IWTYP(IW).EQ.1) THEN
              TYPE 9065
9065          FORMAT ('$Enter the azimuthal angle defining the
     ^ window axis?  ')
              READ (LDT,*) WPHI(IW)
              WTHET(IW)=0.
            ELSE IF(IWTYP(IW).EQ.5) THEN
              TYPE 9066
9066          FORMAT ('$Enter azimuthal,polar angle defining normal
     ^ to the window plane?  ')
              READ (LDT,*) WPHI(IW),WTHET(IW)
            ELSE
              CONTINUE
            END IF
            TYPE *,' '
            TYPE *,'Select the window form as follows:'
            TYPE *,'      enter 0 to select a rectangular window'
            TYPE *,'            1 to define window from table 1'
            TYPE *,'            2 to define window from table 2'
            TYPE *,'            3 to select a Hann window'
            TYPE *,'            4 to select a Hamming window'
            TYPE *,'            5 to select a raised cosine window'
            TYPE *,'            6 to select a raised cos**2 window'
            TYPE *,'            7 to select a Gaussian window'
            TYPE 9070
9070        FORMAT ('$          8 to select a Taylor window        ?? ')
            READ (LDT,*) IWFRM(IW)
            TYPE *,' '

            IF(IWFRM(IW).EQ.5) THEN
              TYPE 9080
9080          FORMAT ('$Raised cosine window.  Enter fractional height of
     ^ pedestal:  ')
              READ(LDT,*) WPAR(IW)
            END IF
            IF(IWFRM(IW).EQ.6) THEN
              TYPE 9085
9085          FORMAT ('$Raised cos**2 window.  Enter fractional height of
     ^ pedestal:  ')
              READ(LDT,*) WPAR(IW)
            END IF
            IF(IWFRM(IW).EQ.7) THEN
              TYPE *,'Gaussian window selected'
              TYPE 9090
9090          FORMAT ('$Enter factor controlling mainlobe
     ^ width/sidelobe levels?  <2.5>??  ')
              READ (LDT,9092) NC,TEMP
9092          FORMAT (Q,F10.0)
              IF(NC.EQ.0) THEN
```

```
              WPAR(IW) = 2.5
            ELSE
              WPAR(IW) = TEMP
            END IF
          END IF
          IF(IWFRM(IW).EQ.8) THEN
            TYPE *,'Taylor window selected'
            TYPE 9093
 9093       FORMAT ('$Enter Taylor parameter controlling mainlobe
     ^ width/sidelobe levels?  <4.>??  ')
            READ (LDT,9092) NC,TEMP
            IF(NC.EQ.0) THEN
              WPAR(IW) = 4.
            ELSE
              WPAR(IW) = TEMP
            END IF
          END IF

          IF (LDT.EQ.5) THEN
            TYPE *,'Compensate window weights for unequal
     ^ spacing of elements over the window'
            TYPE 9095
 9095       FORMAT ('$aperture?  (NOTE that this option should be used
     ^ with caution!) Y/N?  <N>??  ')
          END IF
          READ (LDT,2215) NC,CHR
          IF(CHR(1:1).EQ.'Y'.OR.CHR(1:1).EQ.'y')IWTYP(IW)=IWTYP(IW)+10
 910    CONTINUE
        EWFLAG = .TRUE.          !indicates a window has been defined
        WFLAG = .TRUE.           !flags window so it will be applied
      END IF
      GOTO 20
**********************************************************************

**********************************************************************
*       STEP 10  Calc steering vector for conventional phase to a plane steering

 10     IF(.NOT.STEP3) THEN
          TYPE *,'Step 3 in which freq is input has not been called'
          IF(LDT.NE.5) STOP
          GOTO 20
        END IF
        TYPE *, 'Phase to a plane beam steering selected'
        TYPE *,'Enter the beam steering direction in the array coord
     ^ system'
        TYPE 10000
10000   FORMAT ('$       - azimuthal,polar angles (deg):  ')
        READ (LDT,*) AZANG,PANG

*       Calculate the steering vector
*       Steering vectors are exp(0,-k*p)  where k is the propagation
*               vector and p the position vector.
        TYPE 10010
10010   FORMAT ('$Use actual(A) or ideal(I) element positions for
     ^ steering? <A>??  ')
        READ (LDT,2215) NC,CHR
        USEACTUAL=.TRUE.
        IF(CHR(1:1).EQ.'I'.OR.CHR(1:1).EQ.'i'.AND.STEP5)USEACTUAL=.FALSE.
        IF(CHR(1:1).EQ.'I'.OR.CHR(1:1).EQ.'i'.AND..NOT.STEP5) THEN
          TYPE *,'Ideal positions have not been input (step 5), so
     ^ actual positions will be used'
        END IF
        IF (USEACTUAL.AND..NOT.STEP4) THEN
          TYPE *,'Element positions have not been input (step 4)'
          IF(LDT.NE.5) STOP
          GOTO 20
        END IF
        STEP10 = .TRUE.
        STEP8 = .FALSE.             !steering vectors calculated not input
        CPS = COSD(PANG)
        SPS = SIND(PANG)
        CAS = COSD(AZANG)
        SAS = SIND(AZANG)

*       Calculation of steering vector
        DO 1040 IE=1,NEL
          IF(USEACTUAL.OR..NOT.STEP5) THEN          !use exact ele pstns
            TC=CMPLX(0.,-WVNO*(X(IE)*CPS*CAS+Y(IE)*CPS*SAS+Z(IE)*SPS))
          ELSE
            TC=CMPLX(0.,-WVNO*(XI(IE)*CPS*CAS+YI(IE)*CPS*SAS+ZI(IE)*SPS))
          END IF
          SW(IE) = CONJG(CEXP(TC))
 1040   CONTINUE
        GOTO 20
**********************************************************************

**********************************************************************
*       Step 11  Beamforming and output of beam amplitude and phase response.

 11     IF(.NOT.STEP4) THEN
          TYPE *,'Element positions have not been input (step 4)'
          IF(LDT.NE.5) STOP
```

```
          GOTO 20
        END IF
        IF(.NOT.STEP3) THEN
          TYPE *,'Freq and sound speed have not been input (step 3)'
          IF(LDT.NE.5) STOP
          GOTO 20
        END IF
        IF(.NOT.STEP8.AND..NOT.STEP10) THEN
          TYPE *,'Beam steering vectors have not been defined in
     ^ steps 8 or 10'
          IF(LDT.NE.5) STOP
          GOTO 20
        END IF

*       Set window weights to unity
        DO 1100 IE=1,NEL
 1100   WWT(IE)=1.

*       Calculation of window weights if required
*       ******************************************************************
        IF(STEP10.AND.WFLAG) THEN                   !window weights to be calculated
          DO 1130 IW = 1,NWIND
            PROJMN = 1000.                             !init values for min,max
            PROJMX = -1000.
*           IWTYP GT 10 indicates that aperture sharing is to be used
            IF(IWTYP(IW).EQ.1 .OR. IWTYP(IW).EQ.5 .OR.
     ^        IWTYP(IW).EQ.11 .OR. IWTYP(IW).EQ.15 ) THEN
              COPW=COSD(WTHET(IW))
              SPW=SIND(WTHET(IW))
              CAW=COSD(WPHI(IW))
              SAW=SIND(WPHI(IW))
            END IF

*           Calculation of vector WV when IWTYP=1 to 4 & 11 TO 14, WV is an
*                axial vector and for IWTYP=5,6,15 and 16, WV is a normal
*                vector. WV is used in calculating the aperture projected on
*                the appropriate line or plane
            WV(1)=0.                                   !init value only
            WV(2)=0.
            WV(3)=0.
            IIW=IWTYP(IW)
            IF(IIW.EQ.1.OR.IIW.EQ.11) THEN          !bearing indep hor wind.
              WV(1)=CAW
              WV(2)=SAW
            ELSE IF(IIW.EQ.2.OR.IWW.EQ.12) THEN                !bearing dep hor window
              WV(1)=SAS                                ! window vec is perp
              WV(2)=CAS                                ! to steering direction
            ELSE IF(IIW.EQ.3.OR.IIW.EQ.13) THEN                !brg indep vert wind
              WV(3)=1.
            ELSE IF(IIW.EQ.4.OR.IIW.EQ.14) THEN                !brg dep vert wind
              WV(1)=-SPS*CAS
              WV(2)=-SPS*SAS
              WV(3)=CPS
            ELSE IF(IIW.EQ.5.OR.IIW.EQ.15) THEN                !brg indep 2-d wind
              WV(1)=COPW*CAW
              WV(2)=COPW*SAW
              WV(3)=SPW
              YW(1)=-SAW                            !y-axis in window plane
              YW(2)=CAW
              YW(3)=0.
            ELSE IF(IIW.EQ.6.OR.IIW.EQ.16) THEN                !brg dep 2-d wind
              WV(1)=CPS*CAS
              WV(2)=CPS*SAS
              WV(3)=SPS
              YW(1)=-SAS                            !y-axis in window plane
              YW(2)=CAS
              YW(3)=0.
            END IF
            IF (IIW.EQ.5.OR.IIW.EQ.15.OR.IIW.EQ.6.OR.IIW.EQ.16) THEN
*             Define x-axis and y-axis in window plane for use in calculating
*                center of mass of array
              XW(1)=YW(1)
              XW(2)=YW(2)
              XW(3)=YW(3)
              CALL VECX3(XW,WV)    !crossing y-axis(XW) into window vector WV=x-axs
*             Calculation of center of mass of array projected on window plane
              XSUM=0.
              YSUM=0.
              DO 1105 IE=1,NEL
                IF(USEACTUAL) THEN
                   XSUM = XSUM + X(IE)*XW(1) + Y(IE)*XW(2) + Z(IE)*XW(3)
                   YSUM = YSUM + X(IE)*YW(1) + Y(IE)*YW(2)
                ELSE
                   XSUM = XSUM+XI(IE)*XW(1)+YI(IE)*XW(2)+ZI(IE)*XW(3)
                   YSUM = YSUM + XI(IE)*YW(1) + YI(IE)*YW(2)
                END IF
 1105         CONTINUE
              XM = XSUM/NEL
              YM = YSUM/NEL
            END IF

*           Calculation of the mean and max aperture for use in calculating
```

```
*              window weights
          DO 1110 IE=1,NEL
            IF(IIW.LE.4.OR.(IIW.GT.10.AND.IIW.LE.14)) THEN !axial window
              IF(USEACTUAL)
     ^          WX(IE) = X(IE)*WV(1) + Y(IE)*WV(2) + Z(IE)*WV(3)
              IF(.NOT.USEACTUAL)
     ^          WX(IE) = XI(IE)*WV(1) + YI(IE)*WV(2)      + ZI(IE)*WV(3)
              IF(WX(IE).GT.PROJMX) PROJMX = WX(IE)
              IF(WX(IE).LT.PROJMN) PROJMN = WX(IE)
            ELSE                              !2-D windows
*             Calc proj of position vectors in window plane re center of mass
              IF(USEACTUAL) THEN
                PX(IE) = X(IE)*XW(1) + Y(IE)*XW(2) + Z(IE)*XW(3)-XM
                PY(IE) = X(IE)*YW(1) + Y(IE)*YW(2)-YM
              ELSE
                PX(IE) = XI(IE)*XW(1) + YI(IE)*XW(2) + ZI(IE)*XW(3)-XM
                PY(IE) = XI(IE)*YW(1) + YI(IE)*YW(2)-YM
              END IF
              WX(IE) = SQRT(PX(IE)**2+PY(IE)**2)
              PROJMN = 0.
              IF(WX(IE).GT.PROJMX) PROJMX = WX(IE)
            END IF
1110      CONTINUE
          IF(IIW.LE.4.OR.(IIW.GT.10.AND.IIW.LE.14)) THEN
            APMEAN = (PROJMX+PROJMN)/2              !median projection
            APTOT = PROJMX-PROJMN                          !tot aperture
            ENEL = 2*APTOT*FREQ/C                          !effective # of elements
          ELSE
            ENEL = 4*PROJMX*FREQ/C
          END IF
          APFAC = (ENEL)/(ENEL+2)           !aperture factor adjusts aperture
                                            !for Hann window so window goes to
                                            !zero 1/2 lambda beyond aperture

*         Calculate the weight to compensate for variation in spatial density
*         of the elements --- calculation of aperture share
          DO 1120 IE=1,NEL !determine element multiplicity & nearest neighbor
            AS(IE) = 1.           !amplitude share initialized to 1.
            IF(IIW.GT.10) THEN    !aperture share to be calculated
              IEMUL(IE)=0.            !initialization
              AX(IE) = WX(IE)+10000    !AX (above X) and BX (below X) used for
              BX(IE) = WX(IE)-10000     !    calc of nearest neighbours
              DO 1115 IE1=1,NEL
                IF(WX(IE1).EQ.WX(IE)) THEN
                  IEMUL(IE)=IEMUL(IE)+1 !# of elements co-located
                ELSE
                  IF(WX(IE1).GT.WX(IE).AND.WX(IE1).LT.AX(IE))
     ^                                                   AX(IE)=WX(IE1)
                  IF(WX(IE1).LT.WX(IE).AND.WX(IE1).GT.BX(IE))
     ^                                                   BX(IE)=WX(IE1)
                END IF
                IF(IE1.EQ.NEL) THEN                      !calculate aperture shr
                  IF (IIW.LE.14) THEN
                    IF(AX(IE).EQ.WX(IE)+10000. .AND.
     ^                          BX(IE).EQ.WX(IE)-10000.)
     ^                AS(IE) = 1./IEMUL(IE)              !all elements co-located
                    IF(AX(IE).EQ.WX(IE)+10000.) THEN
                      AS(IE) = (WX(IE)-BX(IE))/IEMUL(IE)
                    ELSE IF(BX(IE).EQ.WX(IE)-10000) THEN
                      AS(IE) = (AX(IE)-WX(IE))/IEMUL(IE)
                    ELSE
                      AS(IE) = ((AX(IE)-BX(IE))/2.)/IEMUL(IE)
                    END IF
                  ELSE IF (IIW.LE.16) THEN                        !2-D window
                    IF(AX(IE).EQ.WX(IE)+10000. .AND.
     ^                           BX(IE).EQ.WX(IE)-10000.) THEN
                      AS(IE) = 1./IEMUL(IE)              !all elements co-located
                    ELSE IF(AX(IE).EQ.WX(IE)+10000.) THEN
                      BX(IE) = (WX(IE)+BX(IE))/2.
                      AS(IE) = ((2*WX(IE)-BX(IE))**2-BX(IE)**2)
     ^                                                /IEMUL(IE)
                    ELSE IF(BX(IE).EQ.WX(IE)-10000) THEN
                      AX(IE) = (AX(IE)+WX(IE))/2.
                      AS(IE) = AX(IE)**2/IEMUL(IE)
                    ELSE
                      AX(IE) = (WX(IE)+AX(IE))/2.
                      BX(IE) = (WX(IE)+BX(IE))/2.
                      AS(IE) = ((AX(IE)**2-BX(IE))**2)/IEMUL(IE)
                    END IF
                  ELSE
                    STOP 'Window type is incorrect'
                  END IF
                END IF
1115          CONTINUE
            END IF                        !end of calculation of aperture share
1120      CONTINUE

*         Calculate the window weight for each element
*         NOTE that the weights are calculated multiplicatively since two
*         windows can be applied at the same time.
          DO 1125 IE=1,NEL
            IF(IIW.LE.4.OR.(IIW.GT.10.AND.IIW.LE.14)) THEN
```

```
              PROJ = 0.
              IF (APTOT.GT.0.) PROJ = ABS(2*(WX(IE)-APMEAN)/APTOT)
            ELSE
              PROJ = 0.
              IF (PROJMX.GT.0.) PROJ = WX(IE)/PROJMX
            END IF
            IF(IWFRM(IW).EQ.0) THEN
             WWT(IE) = WWT(IE)*AS(IE)              !rectangular window
            ELSE IF(IWFRM(IW).EQ.1) THEN           !window in WTBL1
              PROJ = ABS(PROJ)
             CALL TBINTP(NEWTBL(1),1,1,WTBL,PROJ,RES)
             WWT(IE) = WWT(IE)*RES*AS(IE)
            ELSE IF(IWFRM(IW).EQ.2) THEN           !window in WTBL2
              PROJ = ABS(PROJ)
             CALL TBINTP(NEWTBL(2),1,1,WTBL(1,2),PROJ,RES)
             WWT(IE) = WWT(IE)*RES*AS(IE)
            ELSE IF(IWFRM(IW).EQ.3) THEN           !Hann window
             HANN = 0.5 + 0.5*COS(APFAC*PROJ*PI)
             WWT(IE) = WWT(IE)*HANN*AS(IE)
            ELSE IF(IWFRM(IW).EQ.4) THEN           !Hamming window
             HAMM = 0.54*COS(PROJ*PI)+.46
             WWT(IE) = WWT(IE) * HAMM * AS(IE)
            ELSE IF(IWFRM(IW).EQ.5) THEN           !raised cosine window
             RCOS = (1.-WPAR(IW))*0.5*(1+COS(PROJ*PI))+ WPAR(IW)
             WWT(IE) = WWT(IE) * RCOS * AS(IE)
            ELSE IF(IWFRM(IW).EQ.6) THEN           !raised cos**2 window
             R2COS = (1.-WPAR(IW))*COS(PROJ*PI/2)**2+ WPAR(IW)
             WWT(IE) = WWT(IE) * R2COS * AS(IE)
            ELSE IF(IWFRM(IW).EQ.7) THEN           !Gaussian window
             WWT(IE) = WWT(IE)*AS(IE)* EXP(-0.5*(WPAR(IW)*PROJ)**2)
            ELSE IF(IWFRM(IW).EQ.8) THEN           !Taylor window
             WWT(IE) = WWT(IE)*AS(IE)*
     ^                         FIZERO(WPAR(IW)*SQRT(1.-PROJ**2))
            END IF
 1125     CONTINUE
 1130   CONTINUE
      END IF                                        !end of window wt calcs
*     *************************************************************

      IF (.NOT.STEP12.AND..NOT.STEP11) THEN                   !open default o/p files
        OPEN(UNIT=LO1,NAME='BMPAT.AMP',TYPE='NEW',FORM='FORMATTED')
        OPEN(UNIT=LO2,NAME='BMPAT.PHS',TYPE='NEW',FORM='FORMATTED')
      END IF
      IF (STEP12.OR..NOT.STEP11) THEN
*       Output of initial plotting information - these commands included once
*         in each output file
        WRITE (LO1,11000) R1,R2                               !plot origin
        WRITE (LO2,11000) R1,R2
11000   FORMAT (' ORSET ',2(F7.2,1X))
        WRITE (LO1,11010) R3,R4
        WRITE (LO2,11010) R3,R4
11010   FORMAT (' AXSET ',2(F7.2,1X))                                  !axes lengths
        WRITE (LO1,11020) R5
        WRITE (LO2,11020) R5
11020   FORMAT (' THSET ',F7.2)                               !plot angle
*       Calculation of character scaling factor
        IF(R3.LT.R4) THEN
          CWDTH = 100.
          CHGHT = 100.*R3/R4
        ELSE IF(R3.GT.R4) THEN
          CHGHT = 100.
          CWDTH = 100.*R4/R3
        ELSE
          CWDTH = 100.
          CHGHT = 100.
        END IF
        STEP12 = .FALSE.
      ELSE
        WRITE (LO1,*) ' NEWPAGE'
        WRITE (LO1,*) ' PLOT'
        WRITE (LO2,*) ' NEWPAGE'
        WRITE (LO2,*) ' PLOT'
      END IF

      STEP11 = .TRUE.
      LOGT = .TRUE.               !LOGT is to allow only one error msg out

*     Define the plane in which beam is to be sampled as the x-y plane in
*     a beam coordinate system.  This beam system is defined relative the
*     array coord system by three coordinate rotations Z1ROT,YROT,Z2ROT.
*     To describe this transformation it is assumed that the array coord
*     system is such that the x-axis points north, the y-axis
*     east and the z-axis vertically up.  Both the array and the beam systems
*     are left-handed coord systems.  The rotations to go from the array
*     system to the beam coord system are as follows:
*             Z1ROT, YROT are rotations about the z-array and y'-axis which
*                     define the x-y plane of the beam system;
*             Z2ROT is a rotation about the z-axis of the beam system to
*                     orient the x and y axis in the plane.

       TYPE *,'Is beam to be sampled in the horizontal(H)'
      TYPE *,'                                vertical(V)'
```

```
      TYPE 11030
11030 FORMAT ('$                                      or other(O) plane?
    ^ <H>?? ')
      READ (LDT,2215) NC,CHR
      IF(CHR(1:1).EQ.'V'.OR.CHR(1:1).EQ.'v') THEN   !vertical smp plane
        TYPE 11032
11032   FORMAT ('$Azimuthal angle defining orientation
    ^ of vertical sample plane? <0>?? ')
        READ (LDT,11035) NC,VAZANG
11035   FORMAT (Q,F10.0)
        IF (NC.EQ.0) VAZANG=0
        Z1ROT = VAZANG - 90.
        YROT = +90.
        Z2ROT = +90.
        WRITE (LO1,*) ' LABEL 3'
        WRITE (LO1,11040) VAZANG
11040   FORMAT (' !SCA0.6;AMPLITUDE RESPONSE IN VERTICAL PLANE
    ^ BEARING ',F5.0,' degrees')
        WRITE (LO2,*) ' LABEL 3'
        WRITE (LO2,11050) VAZANG
11050   FORMAT (' !SCA0.6;PHASE RESPONSE IN VERTICAL PLANE
    ^ BEARING ',F5.0,' degrees')

      ELSE IF(CHR(1:1).EQ.'O'.OR.CHR(1:1).EQ.'o') THEN  !smp plane not h or v
        TYPE 11055
11055   FORMAT ('$Enter the z,y'' rotations defining the sample
    ^ plane? ')
        READ (LDT,*) Z1ROT,YROT
*       The operator is prompted if the plane in which the beam is to be
*       sampled in not horizontal or vertical.  A YROT greater than 85 degrees
*       is assumed to define a vertical plane.  Prompts only occur if input is
*       from the keyboard.
        IF(LDT.EQ.5.AND.ABS(YROT).LE.85.0) THEN
          Z2ROT = 0.
          CALL XFMROT(Z1ROT,YROT,Z2ROT)
          TYPE *,'With these z,y'' rotations the z" rotation
    ^ required to put the beam x-axis'
          TYPE 11060, Z2ROT
11060     FORMAT ('    in the north plane is ',F6.1,' degrees')
        END IF
        TYPE 11070
11070   FORMAT ('$Enter z" rotation (orients x-axis of the beam
    ^ system in the x-y plane)? ')
        READ (LDT,11080) NC,Z2ROT
11080   FORMAT (Q,F10.0)
        IF(NC.EQ.0) Z2ROT=0.
        WRITE (LO1,*) ' LABEL 3'
        WRITE (LO1,11090) Z1ROT,YROT,Z2ROT
11090   FORMAT (' !SCA0.6;AMPLITUDE RESPONSE IN PLANE ',
    ^                              F4.0,',',F4.0,',',F4.0)
        WRITE (LO2,*) ' LABEL 3'
        WRITE (LO2,11100) Z1ROT,YROT,Z2ROT
11100   FORMAT (' !SCA0.6;PHASE RESPONSE IN PLANE ',
    ^                              F4.0,',',F4.0,',',F4.0)
      ELSE                                              !smp plane horizontal
        TYPE 11110
11110   FORMAT ('$Enter azimuthal angle to which to reference beam
    ^ response? ')
        READ (LDT,*) Z2ROT
        Z1ROT = 0.
        YROT = 0.
        WRITE (LO1,*) ' LABEL 3'
        WRITE (LO1,11120) Z2ROT
11120   FORMAT (' !SCA0.6;AMPLITUDE RESPONSE IN HORIZONTAL PLANE
    ^ relative ',F6.0,' degrees')
        WRITE (LO2,*) ' LABEL 3'
        WRITE (LO2,11130) Z2ROT
11130   FORMAT (' !SCA0.6;PHASE RESPONSE IN HORIZONTAL PLANE
    ^ relative ',F6.0,' degrees')
      END IF

*     Transform element positions into beam ref system
      DO 1135 IE=1,NEL
        XT=X(IE)
        YT=Y(IE)
        ZT=Z(IE)
        CALL XFMZYZ(Z1ROT,YROT,Z2ROT,XT,YT,ZT)
        XB(IE)=XT
        YB(IE)=YT
        ZB(IE)=ZT
 1135 CONTINUE

*     Define the angles in the x-y plane at which the beam is to be sampled.
      TYPE 11140
11140 FORMAT ('$Enter angles at which beam is to be sampled:
    ^  min,max,increment??  ')
      READ (LDT,*) ANGMN,ANGMX,ANGINC
      NB=(ANGMX-ANGMN)/ANGINC+1
      IF(NB.GT.MXNB) THEN
        TYPE *,'The number of samples of beam pattern exceeds array
    ^ dimensions'
        TYPE *,'The angle increment has been increased to stay within
```

```
     ^ dimensions'
          ANGINC = (ANGMX-ANGMN)/(MXNB-1)
          NB = MXNB
        END IF

        TYPE *,'Enter the type of normalization to employ where:'
        TYPE *,' 1. sets the peak response to 0 dB,'
        TYPE *,' 2. normalizes relative to the beam response in
     ^ the steering direction,'
        TYPE *,' 3. normalizes relative to the 3-dimensional peak
     ^ (NOTE this is slow)'
        TYPE 11150
11150   FORMAT ('$
     ^     1,2 or 3? <1>?? ')
        READ (LDT,11151) NC,NORM
11151   FORMAT (Q,I10)
        IF (NC.EQ.0) NORM = 1
        IF (NORM.EQ.2 .AND. STEP8) THEN
          TYPE *,'Normalization 2 cannot be used when steering vector
     ^ is input'
          TYPE *,'    NORM is being changed to 1'
          NORM = 1
        END IF
        IF (NORM.EQ.2.OR.NORM.EQ.3) THEN
          TYPE 11160
11160     FORMAT ('$  Use response of actual(A) or ideal(I) array
     ^ for normalization? <A>?? ')
          READ (LDT,2215) NC,CHR
          IDEAL = .FALSE.
          IF(CHR(1:1).EQ.'I'.OR.CHR(1:1).EQ.'i') IDEAL = .TRUE.
        END IF

        TYPE 11170
11170   FORMAT ('$Is directivity index to be calculated - THIS IS
     ^ TIME CONSUMING - (Y/N)? <N>?? ')
        READ (LDT,2215) NC,CHR
        CALDIR = .FALSE.
        IF(CHR(1:1).EQ.'Y'.OR.CHR(1:1).EQ.'y') CALDIR = .TRUE.

        IF(NORM.EQ.3.OR.CALDIR) THEN
*       Calculation of 3-d norm and directivity index*****************
          TYPE 11180
11180     FORMAT ('$Enter angular step size for peak search/
     ^ directivity calculation? <0.5>?? ')
          READ (LDT,11080) NC,ASTEP
          IF(NC.EQ.0) ASTEP = 0.5
*         Calculate 3-dimensional beam normalization factor and/or directivity
*         index.  Both of these operations are performed by calculating
*         beam response in increments of ASTEP in polar and azimuthal
*         angle.  The element amplitude and phase errors and array distortion
*         are automatically included in the calculations for directivity index
*         and optionally included in the determination of peak response.
          SUMO = 0.                                      !omni power sum
          SUMB = 0.                                      !beam power sum
          BMMX1 = -1000.                    !initial values for max powers
          BMMX2 = -1000.
          NSTEP = 180./ASTEP
          DO 1150 IP = 0,NSTEP
            PA = (IP*ASTEP-90.)
            CP = COSD(PA)
            SP = SIND(PA)
            DO 1145 IA = -NSTEP,NSTEP-1
              AZA = IA*ASTEP
              CA = COSD(AZA)
              SA = SIND(AZA)
              XK = -WVNO*CP*CA
              YK = -WVNO*CP*SA
              ZK = -WVNO*SP
              CPW1 = CMPLX(0.,0.)
              CPW2 = CMPLX(0.,0.)
              LOGT = .TRUE.                 !LOGT is to allow only one error msg out
              DO 1140 IE=1,NEL
*               Calculate element amplitude and phase factor
                IF(IETYP(IE).EQ.0) THEN                    !omni element
                  EAF = 1.
                  EPF = 0.
                ELSE      !calculate angle between element axis and wavevector
                  ELCP = COSD(EEPOL(IE))
                  ELSP = SIND(EEPOL(IE))
                  ELCA = COSD(EEAZ(IE))
                  ELSA = SIND(EEAZ(IE))
                  XT = ELCP*ELCA
                  YT = ELCP*ELSA
                  ZT = ELSP
                  IF ((XT.EQ.0. .AND. YT.EQ.0. .AND. ZT.EQ.0.) .OR.
     ^                (XK.EQ.0. .AND. YK.EQ.0. .AND. ZK.EQ.0.)) THEN
                    ELANG = 0.
                  ELSE
                    ELANG = ACOS( -(XT*XK+YT*YK+ZT*ZK)
     ^                     /SQRT(XT**2+YT**2+ZT**2)
     ^                      /SQRT(XK**2+YK**2+ZK**2))
                    ELANG = ELANG/DTR
```

```
                END IF
             END IF
             IF (IETYP(IE).EQ.1) THEN   !response fm tbl
                IF(.NOT.STEP4) THEN              !element tbl has not been i/p
                  IF(LOGT) THEN          !type out error msg
                    LOGT=.FALSE.               !inhibit error msg
                    TYPE *,'Element table has not been input so ',
     ^                        'type 1 elements assumed omni'
                  END IF
                  EAF=1.
                  EPF=0.
                ELSE
                  CALL TBINTP(NELTB,2,1,ELETBL,ELANG,EAF) !table interpol
                  EAF = 10.**(EAF/20.)
                  CALL TBINTP(NELTB,2,2,ELETBL,ELANG,EPF) !table interpol
                END IF
             ELSE IF (IETYP(IE).EQ.2) THEN                        !dipole
                EAF = ABS(COSD(ELANG))
                EPF = 0.
                IF(COSD(ELANG).LT.0) EPF = 180.
             ELSE IF (IETYP(IE).EQ.3) THEN                        !cardioid
                EPF = 0.
                EAF = 0.5 + 0.5*COSD(ELANG)
             END IF
              IF (NORM.EQ.3.AND.IDEAL) THEN
               PHASE = XI(IE)*XK +YI(IE)*YK + ZI(IE)*ZK + EPF*DTR
               CE = CMPLX(0.,PHASE)
               CPW1 = CPW1 + EAF*WWT(IE)*SW(IE)*CEXP(CE)
              END IF
              IF ((NORM.EQ.3.AND..NOT.IDEAL).OR.CALDIR) THEN
               PHASE = X(IE)*XK +Y(IE)*YK + Z(IE)*ZK + EPF*DTR
     ^                                            + PE(IE)*DTR
                CE = CMPLX(0.,PHASE)
               CPW2 = CPW2 + AE(IE)*EAF*WWT(IE)*SW(IE)*CEXP(CE)
              END IF
1140       CONTINUE
           IF(NORM.EQ.3.AND.IDEAL) THEN
              BP = REAL(CPW1*CONJG(CPW1))
              IF(BP.GT.BMMX1) BMMX1 = BP
           END IF
           IF((NORM.EQ.3.AND..NOT.IDEAL).OR.CALDIR) THEN
              BP = REAL(CPW2*CONJG(CPW2))
              IF(NORM.EQ.3.AND.BP.GT.BMMX1) BMMX1 = BP
              IF(CALDIR)  THEN
                IF(BP.GT.BMMX2) BMMX2 = BP       !beam max for directivity calcs
*               In integrating the elements must be scaled by the size of the
*                      element of solid angle which is proportional to the
*                      cosine of the polar angle
                SUMO = SUMO + CP                 !integrating over omni
                SUMB = SUMB + CP*BP              !integrating over beam
              END IF
           END IF
1145     CONTINUE
1150   CONTINUE
       IF(CALDIR) THEN
         IF(BMMX2.NE.0.0 .AND. SUMO.NE.0.0) THEN
           ARG = SUMB/SUMO/BMMX2
         ELSE
           TYPE *,'There is a problem in integrating for directivity'
           STOP
         END IF
         IF(ARG.LE.10E-30) THEN
           DIRINDX = -1000.
         ELSE
           DIRINDX = -10.*LOG10(ARG)
         END IF
       END IF
       IF(NORM.EQ.3) THEN
         SNORM = SQRT(BMMX1)
         FNORM = 10.*LOG10(MAX(BMMX1,1.E-35))
       END IF
      END IF
*     End of 3-d norm and directivity index calcs*****************
      IF(NORM.NE.3) THEN
        SNORM=1.
        FNORM = 0.
      END IF

*     Calculate and output beam responses
      PASS1 = .FALSE.
      IF(NORM.EQ.2.AND.STEP10) PASS1=.TRUE.  !normalizing factor found on ps 1
      BMMN = 10000.                          !for normalizing in steering dir
      BMMX = -10000.
      DO 1165 IB = 1,NB
1155    IF(PASS1) THEN    !normalizing factor to be calc in steering direction
          CP = COSD(PANG)
          SP = SIND(PANG)
          CA = COSD(AZANG)
          SA = SIND(AZANG)
        ELSE
          ANG = ANGMN+ANGINC*(IB-1)
          CA = COSD(ANG)
```

```
          SA = SIND(ANG)
          CP = 1.
          SP = 0.
        END IF
        XK = -WVNO*CP*CA
        YK = -WVNO*CP*SA
        ZK = -WVNO*SP
        CPW = CMPLX(0.,0.)

        DO 1160 IE = 1,NEL
*         Calculate element amplitude and phase factor which is a function
*         of the type of element, its orientation and the arrival direction
*         of the incoming plane wave.
          IF(IETYP(IE).EQ.0) THEN                               !omni element
            EAF = 1.
            EPF = 0.
          ELSE !calculate angle between element axis and wavevector
            ELCP = COSD(EEPOL(IE))
            ELSP = SIND(EEPOL(IE))
            ELCA = COSD(EEAZ(IE))
            ELSA = SIND(EEAZ(IE))
            XT = ELCP*ELCA
            YT = ELCP*ELSA
            ZT = ELSP
*           Now transform this vector into the beam coord system
            IF(.NOT.PASS1) CALL XFMZYZ(Z1ROT,YROT,Z2ROT,XT,YT,ZT)
*           Calc angle between element axis and propagation vector
            IF ((XT.EQ.0. .AND. YT.EQ.0. .AND. ZT.EQ.0.) .OR.
     ^          (XK.EQ.0. .AND. YK.EQ.0. .AND. ZK.EQ.0.)) THEN
              ELANG = 0.
            ELSE
              ELANG = ACOS( -(XT*XK+YT*YK+ZT*ZK)
     ^               /SQRT(XT**2+YT**2+ZT**2)/SQRT(XK**2+YK**2) )
              ELANG = ELANG/DTR
            END IF
          END IF

          IF (IETYP(IE).EQ.1) THEN                !response fm tbl
            IF(.NOT.STEP6) THEN           !element tbl has not been i/p
              IF(LOGT) THEN               !type out error msg
                LOGT=.FALSE.              !inhibit error msg
                TYPE *,'Element table has not been input so ',
     ^                 'type 1 elements assumed omni'
              END IF
              EAF=1.
              EPF=0.
            ELSE
              CALL TBINTP(NELTB,2,1,ELETBL,ELANG,EAF)     !table interpol
              CALL TBINTP(NELTB,2,2,ELETBL,ELANG,EPF)     !table interpol
              EAF = 10.**(EAF/20.)
            END IF
          ELSE IF (IETYP(IE).EQ.2) THEN                   !dipole
            EAF = ABS(COSD(ELANG))
            EPF = 0.
            IF(COSD(ELANG).LT.0) EPF = 180.
          ELSE IF (IETYP(IE).EQ.3) THEN                   !cardioid
            EPF = 0.
            EAF = 0.5 +0.5* COSD(ELANG)
          END IF
          IF(PASS1) THEN  !calc for normaliz. factor
            IF(IDEAL) THEN
              PHASE = XI(IE)*XK +YI(IE)*YK + ZI(IE)*ZK +EPF*DTR
              CE = CMPLX(0.,PHASE)
             CPW = CPW + EAF*WWT(IE)*SW(IE)*CEXP(CE)
            ELSE
             PHASE = X(IE)*XK +Y(IE)*YK + Z(IE)*ZK + (EPF+PE(IE))*DTR
             CE = CMPLX(0.,PHASE)
             CPW = CPW + AE(IE)*EAF*WWT(IE)*SW(IE)*CEXP(CE)
            END IF
          ELSE
            PHASE = XB(IE)*XK +YB(IE)*YK +ZB(IE)*ZK +(PE(IE)+EPF)*DTR
            CE = CMPLX(0.,PHASE)
            CPW = CPW+EAF*AE(IE)*WWT(IE)*SW(IE)/SNORM*CEXP(CE)
          END IF
1160    CONTINUE
        BP = MAX(REAL(CPW*CONJG(CPW)),1.E-35)
        IF(PASS1) THEN
          SNORM = SQRT(BP)
          FNORM = -10.*LOG10(MAX(BP,1.E-35))      !normalizing factor in dB
          PASS1=.FALSE.
          GOTO 1155
        ELSE
          ANGO(IB) = ANG
          BMP(IB) = 10.*LOG10(BP)
          IF(AIMAG(CPW).EQ.0. .AND. REAL(CPW).EQ.0.) THEN
            PHS(IB) = 0.
          ELSE
            PHS(IB) = ATAN2D(AIMAG(CPW),REAL(CPW))
          END IF
          IF(BMP(IB).GT.BMMX) BMMX=BMP(IB)
          IF(BMP(IB).LT.BMMN) BMMN=BMP(IB)
        END IF
```

```
1165     CONTINUE

*        Calculate the beam width*****************
*           Two types of 3 dB down points are defined for the purpose of finding
*             peaks
*                type 1 - beampower is increasing with increasing beam #
*                type 2 - beampower is decreasing with decreasing beam #

         BWIDTH = 10000                              !beamwidth set to large value
         PKANG = 10000                               !initial ang position of peak
         ISTRT = 1                                   !str point for loop
         DWN3 = BMMX - 3                             !3 dB down from max

*        Find a type 1 3 dB down point
 1170    DO 1175  IB = ISTRT,NB-2
           IF (BMP(IB).LE.DWN3 .AND. BMP(IB+1).GT.DWN3) THEN
             I1 = IB
             GOTO 1180
           END IF
 1175    CONTINUE
         GOTO 1190                                   !no more type 1 3 dB dwn points

*        A type 1 3 dB point has been found - is there a peak?
 1180    PEAK = .FALSE.
         DO 1185 IB = I1+1,NB-1
*          First determine if this beam borders a type 2 3 dB down point
           IF (BMP(IB).GT.DWN3 .AND. BMP(IB+1).LE.DWN3) THEN  !type 2 3dB dwn
             I2 = IB
             IF (PEAK) THEN                          !there is a peak, calc beamwidth
               A1 = (ANGO(I1+1)-ANGO(I1))*(DWN3-BMP(I1))
     ^                        /(BMP(I1+1)-BMP(I1))    + ANGO(I1)
               A2 = (ANGO(I2+1)-ANGO(I2))*(DWN3-BMP(I2))
     ^                        /(BMP(I2+1)-BMP(I2))    + ANGO(I2)
               BWTH = A2 - A1
               IF (BWTH.LT.BWIDTH) THEN
                 BWIDTH = BWTH                       !beamwidth defined as min
                 PKANG = PKA                         !position of peak
               END IF
             END IF
             ISTRT = IB+1
             IF (ISTRT.GT.NB-2) GOTO 1190
             GOTO 1170                               !go to look for type 1 3dB down
           END IF

*          Current beam does not border a type 2 3dB down point. Is it a peak?
*               A peak is declared if bm power is within .1 dB of max power BMMX
           IF (BMP(IB).GE.(BMMX-0.1) .AND..NOT.PEAK) THEN
             PEAK = .TRUE.
             PKA = ANGO(IB)                                   !peak position
             TPEAK = BMP(IB)
*            Find peak position
             DO 1184 IBB = IB+1,NB
               IF (BMP(IBB).GT.TPEAK) THEN
                 PKA = ANGO(IBB)                              !peak position
                 TPEAK = BMP(IBB)
               ELSE
                 GOTO 1185
               END IF
 1184        CONTINUE
           END IF
 1185    CONTINUE

 1190    CONTINUE
*        End of beamwidth calcs*****************

*        Set up ranges for the plots
         YRNG = BMMX-BMMN
         IF(YRNG.LE.10) THEN
           YSTRT = -10
         ELSE IF(YRNG.LE.16) THEN
           YSTRT = -16
         ELSE IF(YRNG.LE.25) THEN
           YSTRT = -25
         ELSE IF(YRNG.LE.40) THEN
           YSTRT = -40
         ELSE IF(YRNG.LE.60) THEN
           YSTRT = -60
         ELSE
           YSTRT = -100
         END IF
         IF(LDT.EQ.5) THEN
           IYSTRT=YSTRT
           TYPE 11190,IYSTRT
11190      FORMAT ('$Enter the minimum beam power to plot:  <',I4,'>??  ')
           READ (LDT,11200) NC,IYSTRT
11200      FORMAT (Q,I10)
           IF(NC.NE.0) YSTRT=IYSTRT
         END IF
         IF(YSTRT.GT.-20) THEN
           DIF=2.
         ELSE IF(YSTRT.GE.-50) THEN
           DIF=5.
```

```
         ELSE
           DIF = 10.
         END IF
         IF (YSTRT.GT.-20) THEN
           YAINC = 2
           NYSBD = 4
         ELSE IF(YSTRT.GE.-35) THEN
           YAINC = 5
           NYSBD = 5
         ELSE
           YAINC = 10
           NYSBD = 5
         END IF
         YMAX=DIF
         YMIN=YSTRT-DIF
         XRNG = ANGMX-ANGMN

         WRITE (LO1,11210) CWDTH,CHGHT
         WRITE (LO2,11210) CWDTH,CHGHT
11210    FORMAT (' CHSET ',2(F7.2,1X))                             !char size
         WRITE (LO1,11220)
         WRITE (LO2,11220)
11220   FORMAT (' SETSPD  1  100')                         !100 mm/sec
         IF(XRNG.EQ.90. .OR.XRNG.EQ.180. .OR.X NG.EQ.360.) THEN
           IF(XRNG.EQ.360.) THEN
             XAINC = 90
             NXSBD = 3
           ELSE
             XAINC = 30
             NXSBD = 6
           END IF
           XSTRT = ANGMN
           IF(XRNG.EQ.90.) THEN
             XAINC = 30
             XMIN = ANGMN-5
             XMAX = ANGMX+5
           ELSE
             XMIN = ANGMN-10
             XMAX = ANGMX+10
           END IF
           WRITE (LO1,11230) 1,XMIN,XMAX
           WRITE (LO2,11230) 1,XMIN,XMAX
           WRITE (LO1,11230) 3,XMIN,XMAX
           WRITE (LO2,11230) 3,XMIN,XMAX
11230     FORMAT (' RANGE',I2,2F8.2)
           WRITE (LO1,11240) 1,XSTRT,XAINC,NXSBD
           WRITE (LO2,11240) 1,XSTRT,XAINC,NXSBD
           WRITE (LO1,11240) 3,XSTRT,XAINC,NXSBD
           WRITE (LO2,11240) 3,XSTRT,XAINC,NXSBD
           WRITE (LO1,*) ' NSET  3  0'
           WRITE (LO2,*) ' NSET  3  0'
         END IF
11240   FORMAT (' SSET ',I2,2F8.2,I5)
         WRITE (LO1,11230) 2,YMIN,YMAX                             !range
         WRITE (LO2,11230) 2,-200.,200.
         WRITE (LO1,11240) 2,YSTRT,YAINC,NYSBD                     !sset
         WRITE (LO2,11240) 2,-180.,30.,6
         WRITE (LO1,11230) 4,YMIN,YMAX                             !range
         WRITE (LO2,11230) 4,-20.,380.
         WRITE (LO1,11240) 4,YSTRT,YAINC,NYSBD                     !sset
         WRITE (LO2,11240) 4,0.,30.,6
         WRITE (LO1,*) ' NSET  4  0'
         WRITE (LO2,*) ' NSET  4  0'

*        Output of information header
         NPLOT = NPLOT +1                          !plot # is incremented
         DI = -100000
         IF (CALDIR) DI = DIRINDX
         NW = NWIND
         IF (.NOT.(STEP10.AND.WFLAG)) NW=0

         IF (NORM.EQ.1) FNORM = -BMX
         WRITE (LO1,*) ' LABEL 7'
         WRITE (LO2,*) ' LABEL 7'
         IF (NW.EQ.0) THEN
           WRITE (LO1,11250) NORM,FNORM,NW
11250      FORMAT (' NORM=',I1,',NORM FACTOR=',F5.1,
     *                 'dB,  WINDOW(S)   #=',I1
           WRITE (LO2,11251) NW
11251      FORMAT (' WINDOW(S)   #=',I1)
         ELSE
           WRITE (LO1,11252) NORM,FNORM,NW,(IWTYP(IW),IWFRM(IW),IW=1,NW
           WRITE (LO2,11253) NW,(IWTYP(IW),IWFRM(IW),IW=1,NW)
11252      FORMAT(' NORM=',I1,',NORM FACTOR=',F5.1,'dB,  WINDOW(S   #=',
     *                     I1,'  type,form',5('  ',I2,',',I1))
11253      FORMAT(' WINDOW(S)   #=',I1,'  type,form',5('  ',I2,',',I1
         END IF
         CALL ERRSET (63,,.FALSE.,,.FALSE.)       !inhibit output conversion err msg
         WRITE (LO1,*) ' LABEL 11'
         WRITE (LO2,*) ' LABEL 11'
         WRITE (LO1,11254) NPLOT,FREQ,BWIDTH,PRANG,DI
         WRITE (LO2,11256) NPLOT,FREQ
```

```
11254   FORMAT (' PLOT #',I2,', FREQ=',F6.0,'Hz,  BMWIDTH=',F5.1,
     -          'deg at ',F5.0,'deg, DI=',F5.1,' dB!SCA1.667;')
11256   FORMAT (' PLOT #',I2,', FREQ=',F6.0,'Hz'SCA1.667;')
        CALL ERRSET (63,,.TRUE.,,.TRUE.,)   !enable output convers error msg

        WRITE (LO1,*) ' LABEL 1'
        WRITE (LO1,*) ' ANGLE (DEGREES)'
        WRITE (LO1,*) ' LABEL 2'
        WRITE (LO1,*) ' BEAM RESPONSE (dB)'
        WRITE (LO2,*) ' LABEL 1'
        WRITE (LO2,*) ' ANGLE (deg)'
        WRITE (LO2,*) ' LABEL 2'
        WRITE (LO2,*) ' BEAM PHASE (deg)'

        WRITE (LO1,*) ' CURVE'
        WRITE (LO2,*) ' CURVE'
        DO 1195 IB=1,NB
          IF(NORM.EQ.1) BMP(IB)=BMP(IB)-BMPMX          !max plotted value set to 0 dB
          WRITE (LO1,*) ANGO(IB),BMP(IB)
          WRITE (LO2,*) ANGO(IB),PHS(IB)
 1195   CONTINUE
        GOTO 20
*****************************************************************

*****************************************************************
*       Step 12  Open output files

 12     IF (STEP12.OR.STEP11) THEN
          CLOSE (UNIT = LO1)
          CLOSE (UNIT = LO2)
        END IF
 1200   TYPE 12000
12000   FORMAT ('$Enter name for output files (include no extension)? ')
        READ (LDT,2000) NC,NAME
        NAME(NC+1) = '.'
        NAME (NC+2) = 'A'
        NAME (NC+3) = 'M'
        NAME (NC+4) = 'P'
        NAME (NC+5) = "C
        OPEN(UNIT=LO1,NAME=NAME,TYPE='NEW',FORM='FORMATTED',ERR=1210)
        NAME(NC+1) = '.'
        NAME (NC+2) = 'P'
        NAME (NC+3) = 'H'
        NAME (NC+4) = 'S'
        OPEN(UNIT=LO2,NAME=NAME,TYPE='NEW',FORM='FORMATTED',ERR=1210)

        STEP12 = .TRUE.
        GOTO 20

 1210   IF(TTYINP) THEN                       !last input was from TTY
          TYPE *,'Error in opening output data files.  TRY AGAIN'
          GOTO 1200
        ELSE                                  !last input was from a file
          STOP 'Error in opening output data file'
        END IF

*****************************************************************

*****************************************************************
*       STEP 13 Exit
 13     CLOSE (UNIT=LO1)
        CLOSE (UNIT=LO2)
        IF(LDT.NE.5) CLOSE(UNIT=LDT)
        STOP
*****************************************************************
        END
```

```
*********************************************************************
*********************************************************************
        SUBROUTINE OPNFIL(LUN,PROMPT,LDT,ERR)

*  ON ENTRY
*     LUN is the logical unit # of file to be opened
*     PROMPT is the character string to be used to prompt for file name.  The
*        character string must be terminated with a '!'
*     LDT is the logical unit # from which file name is to be input

*  ON EXIT
*     LUN,PROMPT and LDT are unchanged
*     ERR is a logical variable that is true if there is an error in opening
*        the file
*********************************************************************

        LOGICAL ERR
        BYTE NAME(32),PROMPT(80)

        IX=0
        DO 10 I=1,80
          IF(PROMPT(I).EQ.'!') GOTO 20
          IX = IX + 1
 10     CONTINUE
 20     CONTINUE
        IF(LDT.EQ.5) TYPE 1000,(PROMPT(J),J=1,IX)
 1000   FORMAT ('$',80A1)
        READ (LDT,1010) NC,NAME
 1010   FORMAT (Q,32A1)
        NAME (NC+1)="0
        ERR = .FALSE.
        OPEN (UNIT=LUN,NAME=NAME,TYPE='OLD',FORM='FORMATTED',ERR=30)
        RETURN

 30     ERR = .TRUE.                        !error opening file
        RETURN
        END

**************************************************************
**************************************************************
        SUBROUTINE VECX3(V1,V2)

*       The subroutine calculates the vector cross product V1 X V2 where
*               V1 and V2 are 3rd order vectors.
*
*       ON EXIT the resultant is in V1

        REAL*4   V1(3),V2(3),VT(3)

        VT(1) = V1(2)*V2(3)-V2(2)*V1(3)
        VT(2) = V1(3)*V2(1)-V2(3)*V1(1)
        VT(3) = V1(1)*V2(2)-V2(1)*V1(2)

        DO 10 IE = 1,3
 10     V1(IE) = VT(IE)

        RETURN
        END
```

```
************************************************************
************************************************************
        SUBROUTINE TBINTP(NENT,NDV,ISEL,TABLE,ARG,RES)

*       A. Collier 14 July/86
*       This subroutine is for interpolating from a table.  The independent
*               values in the table must increase monotonically.
*               The table can contain any number of dependent values.
*       The sequence of values in the table is as follows:
*               indep value(1), dependent1(1),dependent2(1),......
*               indep value(2), dependent1(2),dependent2(2),......
*                       .
*                       .
*                       .
*       ON ENTRY
*               NENT - The number of entries in the table
*               NDV  - The number in dependent variables
*               ISEL - The index defining the dependent variable of interest
*               TABLE: The array containing the table
*               ARG  - The independent variable for which interpolation to be
*                                     made
*       ON EXIT
*               NENT,NDV,ISEL,TABLE,ARG remain unchanged
*               RES  Contains the interpolated value
*
**********************************************************************

        REAL*4   TABLE(*)

*       If ARG is outside the range of the table the returned result is either
*               the first or last table entry
        IF(ARG.LE.TABLE(1)) THEN
          RES = TABLE(ISEL+1)
          RETURN
        END IF
        IX = (NENT-1)*(NDV+1)+1
        IF(ARG.GE.TABLE(IX)) THEN
          RES = TABLE(IX+ISEL)
          RETURN
        END IF

*       Find the position in the table in which ARG falls
        IX2 = 1
        DO 10 IT = 2,NENT
          IX1=IX2
          IX2=IX1+NDV+1
          IF(ARG.GT.TABLE(IX1).AND.ARG.LE.TABLE(IX2)) THEN
            VAR1=TABLE(IX1)
            VAR2=TABLE(IX2)
            DVAR1=TABLE(IX1+ISEL)
            DVAR2=TABLE(IX2+ISEL)
            IF (VAR2.EQ.VAR1) THEN
              TYPE *,'ERROR in table. 2 independent variables are equal'
              STOP
            END IF
            RES = (ARG-VAR1)/(VAR2-VAR1)*(DVAR2-DVAR1)+DVAR1
            RETURN
          END IF
10      CONTINUE
        TYPE *,'An error has occured during table interpolation due to
     ^ improper table values'
        STOP
        END
```

```
**********************************************************************
**********************************************************************
      FUNCTION FIZERO(ARG)

*     Chebychev expansion of I0(ARG) (modified Bessel function type 0)
**********************************************************************

      REAL*4 A(13)/255.46687962436217,190.49432017274284,
     ^              82.48903274402410,22.27481924246223,4.01167376017935,
     ^              0.50949336543998,.04771874879817,.00341633176601,
     ^              .00019246935969,.0000087383155,.00000032609105,
     ^              .00000001016973,.00000000026883/

      FIZERO = SUM (ARG/9.0,A,13)
      RETURN
      END

**********************************************************************
**********************************************************************
      FUNCTION SUM(Z,A,NN)

      REAL*4   A(13)

      N=NN
      B2=0.
      B3=0.
      X=2*Z*Z-1.
1     B1 = 2*X*B2-B3+A(N)

      IF(N.EQ.1) GOTO 2
      B3 = B2
      B2 = B1
      N = N-1
      GOTO 1
2     SUM = (B1-B3)/2.0
      RETURN
      END
```

```
*******************************************************************
*******************************************************************
        SUBROUTINE XFMROT (Z1ROT,YROT,Z2ROT)

*******************************************************************
*  A. Collier 11 Feb/87 version
*
*  Given a left-hand co-ordinate system (positive rotations are in a clockwise
*       direction), referred to as co-ordinate system 1, in which the x-y plane
*       can be imagined to be horizontal, the z-axis vertical, and the
*       x-z plane to be the north plane;  the x-y plane can be oriented in any
*       way by a sequence of two rotations:  The first about the z-axis (Z1ROT)
*       and the second about the resultant y-axis or y'-axis (YROT).
*       These two rotations result in a x",y",z" co-ordinate sytem, which has
*       the x"-y" plane oriented as desired.  However the position of the
*       x" and y" axis in this plane, relative to the north plane is not
*       readily obvious.  This subroutine calculates the z"-axis rotation (Z2ROT)
*       required to put the x"-axis in the north plane.
*
*   The Z2ROT value calculated can be in error when the x"-y" plane is near
*       vertical relative co-ordinate system 1 (this corresponds to YROT values
*       that approach + or - 90 degrees).  Consequently if YROT is outside the
*       range +,- 85 degrees, then an advisory is typed and the subroutine is
*       exited without further calculation.
*
*   ON ENTRY
*       Z1ROT,YROT contain the z-axis, y'-axis rotations (in degrees)  to
*               orient the x-y plane, relative co-ordinate system 1.
*       If YROT is outside the range +,- 85 degrees, and advisory is typed and
*               the routine exited.
*
*   ON EXIT
*       Z1ROT,YROT are unchanged
*       Z2ROT contains the z"-axis rotation require to rotate the x"-axis into
*               the north plane.
*****************************************************************

        DATA DTR/0.017453293/                    !degrees to radians

        Z1R=Z1ROT*DTR
        YR = YROT*DTR
        IF (YROT.GT.85. .OR. YROT.LT.-85.) THEN
          TYPE *,'Subroutine XFMROT,  x"-y" plane is too near vertical,
     ^ EXITED S/R with X2ROT=0'
          RETURN
        END IF

*       Calculate the z"-axis rotation required to put x"-axis in the north
*       plane.

        CSZ1 = COSD(Z1ROT)
        SNZ1 = SIND(Z1ROT)
        CSY = COSD(YROT)
        SNY = SIND(YROT)
        TNY = TAND(YROT)

*       XCLIN  is the angle beteween the x-axis of co-ordinate system 1 and the
*         vertex formed by the intersection of the x"-y" plane and the north
*         plane.
        CSXCLIN= (1.+TNY**2*CSZ1**2)**-.5  !COS(XCLIN)
        SNXCLIN= TNY*CSZ1*CSXCLIN          !SIN(XCLIN)
        SIGN = 1.
        IF(SNZ1.NE.0) SIGN = -SNZ1/ABS(SNZ1)
        Z2ROT = SIGN*ACOS(CSZ1*CSY*CSXCLIN+SNY*SNXCLIN)/DTR
        RETURN

        END
```

```
**********************************************************************
**********************************************************************
      SUBROUTINE XFM2YZ(Z1ROT,YROT,Z2ROT,X,Y,Z)

**********************************************************************
*   A. Collier 11 Feb/87 version
*
*   Given co-ordinate system 1, which can be imagined to have the x-axis
*        pointing north, the y-axis east and the z-axis vertically up;
*
*   the co-ordinates of a point (point A) in co-ordinate system 1 = X,Y,Z
*
*   and the Z,Y,Z rotations (Z1ROT,YROT,Z2ROT) that define a second co-ordinate
*        system relative to system 1, system 2,  (both co-ordinate sytems are
*                 left-handed so positive angles are clock-wise)
*
*   this subroutine calculates the co-ordinates of the point X,Y,Z in co-
*        ordinate system 2 (= X",Y",Z")
*
*   ON ENTRY
*        Z1ROT,YROT,Z2ROT are the Z,Y,Z rotations in degrees that define
*          co-ordinate sytem 2 relative co-ordinate system 1.
*
*        X,Y,Z are the coordinates of point A in the original coordinate system.
*
*   ON EXIT
*        Z1ROT,YROT,Z2ROT are unchanged
*        X,Y,Z are the co-ordinates of point A in co-ordinate system 2
C
C   REQUIRED SUBROUTINES
C        ROTATE  it performs a rotational transformation about a selected axis
C
C*********************************************************************

      REAL*4    V(3)

      V(1)=X
      V(2)=Y
      V(3)=Z
      CALL ROTATE(Z1ROT,3,V)
      CALL ROTATE(YROT,2,V)
      CALL ROTATE(Z2ROT,3,V)
      X=V(1)
      Y=V(2)
      Z=V(3)

      RETURN
      END
```

```
*********************************************************************
*********************************************************************
        SUBROUTINE ROTATE (ANG,IAX,V)

*********************************************************************
*
*       A. Collier 11 Feb/87 version, original 10 Dec/85
*       This subroutine performs a rotational transformation of the vector
*       V to put the vector into a coord system rotated about the IAX axis
*       by an angle ANG (degrees) relative to the original coord system.
*       Coordinate systems are assumed to be left-handed in which  a positive
*       rotation is rotation in a clockwise direction.
*
*       ON ENTRY
*               ANG - The angle in degrees by which the original coord system
*                       rotated to obtain the new system
*               IAX - The axis about which the above rotation takes place
*                       1 indicates x-axis, 2 the y-axis and 3 the z-axis
*               V   - The vector to be transformed into the new system
*
*       ON EXIT
*               ANG and IAX unchanged
*               V   - The transformed vector
*
*********************************************************************

        REAL*4   V(3),VT(3),C(3,3)

        IF(ANG.EQ.0.) RETURN                    !no transformation required
        CSA=COSD(ANG)
        SNA=SIND(ANG)
        C(1,1)=1.                               !rotational coefs for x rotation
        C(1,2)=0.
        C(1,3)=0.
        C(2,1)=0.
        C(2,2)=CSA
        C(2,3)=SNA
        C(3,1)=0.
        C(3,2)=-SNA
        C(3,3)=CSA

        INC=IAX-1                               !index increment
        DO 10 I=1,3
          IX=I-INC
          IF(IX.LE.0) IX=IX+3
          VT(I)=0.                              !initialization
          DO 10 J=1,3
            JX=J-INC
            IF(JX.LE.0) JX=JX+3
            VT(I)=VT(I)+V(J)*C(IX,JX)
10      CONTINUE

        DO 20 I=1,3
20      V(I)=VT(I)

        RETURN
        END
```

UNLIMITED DISTRIBUTION

A18730

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATING ACTIVITY Defence Research Establishment Atlantic	2a. DOCUMENT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. DOCUMENT TITLE BMPAT: A Program for Calculation and Display of the Response of Three-Dimensional Arrays	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)	
5. AUTHOR(S) (Last name, first name, middle initial) A.J. Collier	
6. DOCUMENT DATE April 1987	7a. TOTAL NO. OF PAGES 63 7b. NO. OF REFS 3
8a. PROJECT OR GRANT NO.	9a. ORIGINATOR'S DOCUMENT NUMBER(S) DREA Technical Communication 87/306
8b. CONTRACT NO.	9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)
10. DISTRIBUTION STATEMENT Unlimited Distribution	
11. SUPPLEMENTARY NOTES	12. SPONSORING ACTIVITY

13. ABSTRACT

The program BMPAT is a FORTRAN 77 program designed to readily evaluate the amplitude and phase response of general three-dimensional sonar receiving arrays to plane wave arrivals. The program output is directed to two output files, one for amplitude response and the other for phase response, both of which are formatted for plotting using the DREA Surveillance Acoustics Plotting Package, SAPLOT. Features in the program include the ability to handle array elements that have directional response, provision for element amplitude and phase imbalance and for array distortion. The beamforming operation is a linear operation and the steering vectors can either be input directly or can be calculated for phase-to-a-plane beamforming. There are several amplitude window functions resident in the program and additional functions can be input via tables. Beam patterns are sampled in planes which can be arbitrarily oriented relative to the array co-ordinate system. Beamwidths in the sample planes are calculated and the directivity index of beams can optionally be calculated.

DSIS 78-078

Unclassified

Security Classification

KEY WORDS

Beamforming
Array Response
Windowing
Sonar Arrays

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP: Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER: If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S): Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this document from their defence documentation center."

(2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

The length of the abstract should be limited to 20 single-spaced standard typewritten lines; 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.

END

7-87

DTIC